



UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Maestría en Gestión Estratégica de Tecnologías de la Información, Segunda Cohorte

Análisis del rendimiento de un sistema cooperativo de acceso múltiple ortogonal / no-ortogonal (OMA/NOMA) gestionado mediante una red neuronal artificial.

Trabajo de titulación previo a la obtención del título de Magíster en Gestión Estratégica de Tecnologías de la Información.

Autor :

Ing. Juan Diego Belesaca Mendieta

CI: 0104817770

Correo electrónico: juanodiegobelesacam@gmail.com

Director :

Ing. Andrés Marcelo Vázquez Rodas, PhD

CI: 0301496840

Cuenca - Ecuador

16-marzo-2021



Resumen

Las tecnologías inalámbricas de próxima generación enfrentan desafíos considerables en términos de proporcionar muy baja latencia y alta conectividad, requisitos imprescindibles de las nuevas redes móviles heterogéneas. Impulsado principalmente por estos problemas, este estudio se centra en aumentar la conectividad del usuario, mejorando al mismo tiempo el rendimiento general del sistema. Para esto, se propone y evalúa un sistema híbrido de acceso múltiple ortogonal no-ortogonal impulsado por aprendizaje automático (OMA/NOMA). Específicamente, en este trabajo, se propone el uso de una red neuronal artificial (*Artificial Neural Network ANN*) para seleccionar el método de acceso OMA o NOMA de cada equipo de usuario (*User Equipment UE*). Como parte de esta investigación, también se evalúa la precisión y el tiempo de entrenamiento de los tres algoritmos de aprendizaje más relevantes de las ANN (L-M, BFGS y OSS). El objetivo principal es aumentar el rendimiento de la red móvil de próxima generación, considerando el entorno del canal de onda milimétrica (*millimeter-Wave mm-Wave*) y la formación de haces (*beamforming*).

Los resultados de la evaluación de la propuesta muestran un incremento en el rendimiento promedio del sistema de hasta 20 %, en términos de la tasa total efectiva del sistema, utilizando la administración ANN en contraste con un sistema administrado aleatoriamente sin ANN. El algoritmo de entrenamiento Leveberg-Marquard (L-M) es el mejor algoritmo general para esta aplicación. El mismo presenta la mayor precisión de alrededor de 77 % a costa de un tiempo de entrenamiento de aproximadamente 37 minutos. Sin embargo, sacrificando la precisión hasta un valor más bajo de 73 %, se requiere un tiempo de entrenamiento de aproximadamente 28 segundos.

Palabras clave : NOMA. OMA. Red Neuronal Artificial ANN. Canal mm-Wave. Formación de Haz Analógico. Sum-rate. L-M. BFGS. OSS.



Abstract

Next-generation wireless technologies face considerable challenges in terms of providing the required latency and connectivity for new heterogeneous mobile networks. Driven by these problems, this study focuses on increasing user connectivity together with the system general throughput. For doing so, we propose and evaluate a hybrid machine learning-driven orthogonal/non-orthogonal multiple access (OMA/NOMA) system. Specifically, in this work, we use an artificial neural network (ANN) to assign an OMA or NOMA access method to each user equipment (UE). As part of this research we also evaluate the accuracy and training time of the three most relevant learning algorithms of ANN (L-M, BFGS, and OSS). The main objective is to increase the sum-rate of the next generation mobile network in the current beamforming and millimeter-Wave (mm-Wave) channel environment.

Simulation results show up to a 20% sum-rate average performance increase of the system using the ANN management in contrast to a random non-ANN managed system. The Levenberg-Marquard (L-M) training algorithm is the best overall algorithm for this proposed application as it presents the highest accuracy of around 77% despite 37 minutes of training, and lower accuracy of 73% with approximately 28 seconds of training time.

Keywords : NOMA. OMA. Artificial Neural Networks ANN. mm-Wave Channel. Analog Beamforming. Sum-rate. L-M. BFGS. OSS.



Índice general

Resumen	2
Abstract	3
Índice general	4
Índice de figuras	8
Índice de tablas	10
Dedicatoria	14
Agradecimientos	15
Abreviaciones y acrónimos	16
1. Introducción	18
1.1. Identificación del Problema	19
1.2. Justificación	19
1.3. Alcance	20
1.4. Objetivos	21



1.4.1. Objetivo General	21
1.4.2. Objetivos específicos	21
2. Marco Teórico	23
2.1. Técnicas de Acceso Multiple	24
2.1.1. Acceso Múltiple Ortogonal (OMA)	24
2.1.1.1. Acceso Múltiple por División de Tiempo (TDMA)	24
2.1.1.2. Acceso Múltiple por División de Frecuencia (FDMA)	24
2.1.1.3. Acceso Múltiple por División de Frecuencia Ortogonal (OFDMA)	25
2.1.2. Acceso Múltiple No Ortogonal (NOMA)	26
2.1.2.1. NOMA en el Dominio de la Potencia	27
2.2. Formación de Haces (Beamforming)	29
2.2.1. Beamforming Analógico	29
2.2.2. Beamforming Digital	30
2.2.3. Beamforming Híbrido	30
2.3. Canal de Ondas Milimétricas (mm-Wave)	30
2.4. Red Neuronal Artificial (ANN)	31
2.4.1. Modelo computacional de una ANN	31
2.4.2. Función de Activación	32
2.4.3. Algoritmos de entrenamiento	33
2.4.3.1. Algoritmo de Broyden–Fletcher–Goldfarb–Shanno (BFGS)	34
2.4.3.2. Algoritmo Secante de un Paso (One-Step Secant OSS)	34
2.4.3.3. Algoritmo de Levenberg-Marquardt (L-M)	35



3. Trabajos relacionados	37
4. Metodología	40
4.1. Especificación del Sistema Híbrido OMA/NOMA	41
4.2. Generación del Conjunto de Datos	42
4.3. Sistema de Gestión ANN	45
4.4. Criterios de Evaluación del Desempeño	47
5. Resultados	49
5.1. Análisis de Desempeño de la ANN	50
5.1.1. Análisis del Tiempo de Entrenamiento	50
5.1.2. Análisis de la Precisión del Entrenamiento	51
5.1.3. Análisis de la Cantidad de Iteraciones del Entrenamiento	52
5.1.4. Análisis de la Función de Activación	52
5.2. Análisis del Desempeño del Sistema Híbrido OMA/ NOMA	55
5.2.1. Análisis del Desempeño del Sistema Híbrido OMA/NOMA administrado por la ANN con el Menor Tiempo de Entrenamiento.	55
5.2.2. Análisis del Desempeño del Sistema Híbrido OMA/NOMA administrado por la ANN con Mayor Precisión.	55
5.3. Conclusiones	57
6. Conclusiones	58
6.1. Conclusiones	59
6.2. Trabajos Futuros	59
A. Análisis Previo de Arquitecturas	61



A.1. Resultados del Análisis Previo de Arquitecturas de ANN	62
B. Scripts	64
B.1. Script Generador de Datos	65
B.2. Script Principal	65
Bibliografía	67



Índice de figuras

1.1. Modelo de sistema considerado para el estudio de la coexistencia OMA/NOMA.	21
2.1. Método de acceso TDMA.	25
2.2. Método de acceso FDMA.	25
2.3. Método de acceso OFDMA.	26
2.4. Ejemplo de asignación de potencia a cada UE	27
2.5. Decodificación en un enlace descendente de NOMA	28
2.6. Ejemplo de un esquema básico de una ANN	32
4.1. Comparación entre un sistema con acceso OMA y un sistema con acceso híbrido OMA/NOMA.	42
5.1. Diagrama de caja del número de iteraciones de ANN para 1 HL y para 3 HL.	52
5.2. Resultados del cambio de función de activación (sigmoide y tangente hiperbólica) para la ANN de 20 neuronas en 1 HL.	53
5.3. Resultados del cambio de función de activación (sigmoide y tangente hiperbólica) para la ANN de 50-30-20 neuronas en 3 HL.	54
5.4. Rendimiento del sistema híbrido OMA/NOMA gestionado por una ANN con 20 neuronas en 1 HL.	56



5.5. Rendimiento del sistema híbrido OMA/NOMA gestionado por una ANN con 50-30-20 neuronas en 3 HL.	57
--	----



Índice de tablas

2.1. Funciones de activación más utilizadas.	33
4.1. Combinaciones posibles de UEs	44
4.2. Ejemplo numérico de datos a utilizar en el entrenamiento y/o validación de la ANN.	44
4.3. Arquitecturas ANN seleccionadas	46
5.1. Resultados del tiempo de entrenamiento de la ANN.	50
5.2. Resultados de la precisión de la ANN.	51
5.3. Aumento promedio del rendimiento del sistema híbrido OMA/NOMA.	56
A.1. Análisis previo con el algoritmo de entrenamiento BFGS	62
A.2. Análisis previo con el algoritmo de entrenamiento OSS	63
A.3. Análisis previo con el algoritmo de entrenamiento L-M	63



Cláusula de Propiedad Intelectual

Yo, Juan Diego Belesaca Mendieta autor del trabajo de titulación “Análisis del Rendimiento de un Sistema Cooperativo de Acceso Múltiple Ortogonal/No-Ortogonal (OMA/NOMA) Gestionado Mediante una Red Neuronal Artificial”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 16 de marzo de 2021

A handwritten signature in blue ink, reading "Juan D. Belesaca Mendieta", with a horizontal line underneath.

Juan Diego Belesaca Mendieta

C.I: 0104817770



Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Juan Diego Belesaca Mendieta en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación “Análisis del Rendimiento de un Sistema Cooperativo de Acceso Múltiple Ortogonal/No-Ortogonal (OMA/NOMA) Gestionado Mediante una Red Neuronal Artificial”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 16 de marzo de 2021

Juan Diego Belesaca Mendieta

C.I: 0104817770



CERTIFICO

Que el presente trabajo de titulación: “Análisis del Rendimiento de un Sistema Cooperativo de Acceso Múltiple Ortogonal/No-Ortogonal (OMA/NOMA) Gestionado Mediante una Red Neuronal Artificial”, fue dirigido y revisado por mi persona.

A handwritten signature in blue ink, reading "VÁZQUEZ R", written over a horizontal line.

Ing. Andrés Marcelo Vázquez Rodas, PhD
Director



Dedicatoria

A mis padres,

por ser un pilar fundamental en mi formación académica y personal, por sus consejos, sacrificios y valores inculcados que me hicieron crecer como persona. Por su constante apoyo y motivación una vez mas, que me permitió concluir un objetivo importante de mi vida. Absolutamente todo este trabajo ha sido posible gracias a ellos.

A mi hermana,

la futura ingeniera y colega. Gracias por motivarme a mejorar día con día, por ser una amiga y confidente. Te quiero mucho.

A mis amigos,

que gracias a su ayuda en buenos y malos momentos logré salir adelante con esta nueva etapa en mi formación profesional.

A mi nueva familia,

para mi hijo o hija que esta próximo a llegar y a su madre Jessica, los amo mucho. Este es un paso muy importante que cimienta las bases de un futuro prospero para nosotros juntos.

Ing. Juan Diego Belesaca



Agradecimientos

Al director Ingeniero Andrés Vázquez,

por confiar en nosotros para llevar acabo este trabajo. Gracias por su esfuerzo, dedicación, tutoría, motivación académica y personal como amigo que han sido imprescindibles para el desarrollo de este trabajo.

Al Ingeniero Pablo Avila Campos,

que me brindó ayuda incondicional, con sus amplios conocimientos en NOMA y redes móviles.

¡Gracias a ustedes!

Ing. Juan Diego Belesaca



Abreviaciones y Acrónimos

1G 1st Generation. [19](#)

2G 2nd Generation. [19](#), [24](#)

3G 3rd Generation. [19](#)

4G 4th Generation. [19](#), [26](#)

5G 5th Generation. [19](#)

ANN Artificial Neural Network. [2](#), [3](#), [5–10](#), [20](#), [21](#), [23](#), [31–34](#), [36–39](#), [41](#), [44–47](#), [50–57](#), [59](#), [62](#)

BFGS Broyden–Fletcher–Goldfarb–Shanno. [2](#), [3](#), [5](#), [10](#), [34](#), [39](#), [46](#), [50–53](#), [55](#), [56](#), [59](#), [62](#)

BS Base Estation. [20](#), [27](#), [29](#), [30](#), [38](#), [41](#)

C-IoT Cellular Internet of Things. [19](#)

CDMA Code Division Multiple Access. [19](#), [24](#)

CSI Channel Status Information. [38](#)

D2D Device to Device. [19](#)

FDMA Frequency Division Multiple Access. [5](#), [8](#), [19](#), [24–26](#)

HL Hidden Layer. [8](#), [9](#), [45](#), [46](#), [50–57](#)

L-M Levenberg-Marquardt. [2](#), [3](#), [5](#), [10](#), [35](#), [38](#), [39](#), [46](#), [50–53](#), [55](#), [56](#), [59](#), [63](#)

LOS Line-of-Sight. [30](#), [31](#), [42](#)

LSTM Long Short-Term Memory Network. [38](#)

ML Machine Learning. [38](#), [39](#)

mm-Wave millimeter-Wave. [2](#), [3](#), [5](#), [20](#), [23](#), [30](#), [42](#)

MTC Machine-Type Communication. [19](#)

NLOS Non-Line-of-Sight. [31](#)

NOMA No-Orthogonal Multiple Access. [2](#), [3](#), [5](#), [6](#), [8–10](#), [19–21](#), [23](#), [24](#), [26–29](#), [31](#), [34](#), [37–39](#), [41–44](#), [47](#), [49](#), [55–57](#), [59](#)



- OFDMA** Orthogonal Frequency Division Multiple Access. [5](#), [8](#), [19](#), [24–26](#)
- OMA** Orthogonal Multiple Access. [2](#), [3](#), [5](#), [6](#), [8–10](#), [19–21](#), [23](#), [24](#), [29](#), [34](#), [37–39](#), [41](#), [42](#), [44](#), [47](#), [49](#), [55–57](#), [59](#)
- OSS** One-Step Secant. [2](#), [3](#), [5](#), [10](#), [34](#), [39](#), [46](#), [50–53](#), [55](#), [56](#), [59](#), [63](#)
- RAM** Random Access Memory. [45](#)
- RB** Resource Blocks. [24](#), [26](#)
- SDMA** Space Division Multiple Access. [19](#)
- SIC** Successive Interference Cancellation. [28](#), [29](#), [31](#), [38](#)
- SINR** Signal-to-Interference-Plus-Noise ratio. [28](#)
- SNR** Signal-to-Noise Ratio. [28](#), [29](#), [38](#), [42](#), [48](#), [55](#)
- TDMA** Time Division Multiple Access. [5](#), [8](#), [19](#), [24–26](#)
- UE** User Equipment. [2](#), [3](#), [8](#), [10](#), [20](#), [24–29](#), [31](#), [34](#), [38](#), [39](#), [41–44](#), [49](#), [55–57](#), [60](#)
- URA** Uniform Rectangular Array. [29](#), [41](#)



Capítulo 1

Introducción

En este capítulo se presenta la identificación del problema y justificación del proyecto, así como el alcance y objetivos del mismo.

1.1. Identificación del Problema

El impacto y uso de las comunicaciones móviles a lo largo de estos años ha tenido un crecimiento y una evolución sin precedentes. El tráfico global de datos móviles está creciendo rápidamente, las predicciones indican que el 35 % del tráfico total de datos de Internet se realizará a través de redes 5G para 2024 [1]. En consecuencia, los sistemas de comunicaciones móviles de próxima generación tienen numerosos desafíos que resolver, como la conectividad masiva de dispositivos y el uso eficiente del espectro radioeléctrico.

Las técnicas de acceso múltiple ortogonales OMA tradicionales (Por ejemplo, CDMA en 3G y OFDMA en 4G), se han visto limitadas precisamente por los dos aspectos antes mencionados. Por otro lado, la nueva técnica de Acceso Múltiple No Ortogonal (NOMA) se presenta como una alternativa prometedora para resolver los desafíos antes mencionados [2], debido a que ofrece mayores capacidades de conectividad dentro de los mismos recursos espectrales [3]. Además, la posible combinación de NOMA con diversas técnicas ha demostrado resultados interesantes en estudios recientes (Ej., [4–8]), y generan diferentes desafíos en la estimación de canales, asignación de potencia, convivencia, entre otros.

1.2. Justificación

El impacto y utilización de las comunicaciones móviles a lo largo de estos años ha tenido un crecimiento y evolución sin precedentes. Desde las primeras generaciones de comunicaciones móviles enfocadas en servicios de voz (1G – 2G), hasta el desarrollo actual de quinta generación (5G) en el que, la conmutación de paquetes, el acceso a Internet de ultra banda ancha, la comunicación dispositivo a dispositivo (*Device to Device* D2D), el Internet celular de las cosas (*Cellular Internet of Things* C-IoT), las comunicaciones tipo máquina (*Machine-Type Communication* MTC), las comunicaciones vehiculares, entre otros, constituyen los nuevos retos y paradigmas de la nueva generación de comunicación móvil.

Los métodos de acceso múltiple ortogonal (OMA) convencionales, se han establecido como los métodos de control de acceso al medio fundamentales para las redes móviles de hasta 4G por su simplicidad y eficiencia. Entre estos están el acceso múltiple por división de tiempo (*Time Division Multiple Access* TDMA) y frecuencia (*Frequency Division Multiple Access* FDMA) base para la 2G, el acceso múltiple por división de código (*Code Division Multiple Access* CDMA) base de la 3G, el acceso múltiple por división de frecuencia ortogonal (*Orthogonal Frequency Division Multiple Access* OFDMA) base para la 4G, y el acceso múltiple por división espacial (*Space Division Multiple Access* SDMA), una de las opciones para la 5G.



Por otra parte, **NOMA** representa una nueva técnica de acceso múltiple no ortogonal que utiliza la superposición de código o potencia de diferentes señales para controlar el acceso de los equipos de usuario (*User Equipment UE*). Con la característica relevante de ofrecer mayores capacidades de conectividad dentro de los mismos recursos espectrales [3].

La combinación de **NOMA** con técnicas como: la conformación del haz (*beamforming*), el uso de ondas milimétricas (**mm-Wave**), o la coexistencia con **OMA** (entre otras) han generado un importante interés en la comunidad científica. Los principales esfuerzos por estudiar estos métodos han colocado a estas técnicas como potenciales candidatas para ser usadas en las redes móviles de próxima generación [4–8].

Bajo este contexto, en este proyecto se analiza y evalúa la coexistencia de un sistema híbrido **OMA** y **NOMA**. A través de un primer conjunto de simulaciones se obtienen las características de la red en cuanto al canal, las implicaciones de la conformación del haz y el rendimiento del sistema. Este primer paso permite identificar los casos y condiciones en los que se puede mejorar el rendimiento general de la red en términos de la tasa de transmisión global. Con los datos obtenidos, en una segunda etapa, se entrena una red neuronal, **ANN**, con el objetivo de que una estación base, **BS**, pueda gestionar automáticamente y de manera eficiente los usuarios a los cuales debe dar acceso por **OMA** o **NOMA**. Adicionalmente se estudian diferentes algoritmos de aprendizaje, funciones de activación y número de capas para la **ANN** con el objetivo de evaluar distintas opciones y en base a los resultados definir la combinación que mejor se adapta a la aplicación propuesta en este trabajo.

1.3. Alcance

En este proyecto se trabaja sobre un entorno controlado de una red móvil de quinta generación que consiste en una estación base y tres equipos de usuario (dos controlados mediante **NOMA** y uno mediante **OMA**). Además, se evalúan diferentes algoritmos de una **ANN** que permitan una selección óptima del método de acceso entre **OMA** y **NOMA**. Se asume que la **BS** se dirige hacia los usuarios mediante una conformación de haz analógico de dos haces, usando ondas milimétricas, **mm-Wave**, y un arreglo de antenas de disposición cuadrada con 25 elementos. El haz que utilice **NOMA** será capaz de dar servicio a dos **UE**, mientras que el haz que utilice **OMA** dará servicio al **UE** restante (Figura 1.1).

Las simulaciones se llevarán a cabo utilizando diferentes ángulos de incidencia y variaciones en los canales para cada **UE**. Esto con el fin de dar una generalidad al sistema, y asumiendo diferentes posiciones de los usuarios en el entorno. De los resultados obtenidos de la simulación se seleccionarán los que mejor rendimiento de la red generen. Con estos resultados preliminares

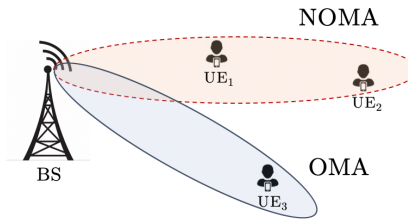


Figura 1.1: Modelo de sistema considerado para el estudio de la coexistencia [OMA](#)/[NOMA](#).

se dispondrá de un conjunto de datos para el entrenamiento, prueba y otro para la validación de la [ANN](#).

Posteriormente, se evalúan tres algoritmos de entrenamiento, diferentes cantidades de capas que conforma la [ANN](#), y se complementa el análisis con dos diferentes funciones de activación de las neuronas de las capas ocultas. Finalmente, se obtendrán gráficas y resultados comparativos en los que se observa la mejor configuración de la [ANN](#) para esta aplicación en particular de coexistencia entre [OMA](#) y [NOMA](#).

1.4. Objetivos

1.4.1. Objetivo General

- Gestionar el acceso eficiente de los usuarios de una red móvil a la estación base mediante técnicas multi-acceso ortogonales y no-ortogonales ([OMA](#) o [NOMA](#)) con el soporte de diferentes algoritmos de redes neuronales artificiales y en base a las características de estado del canal y ángulo de incidencia de los usuarios.

1.4.2. Objetivos específicos

- Evaluar mediante simulación un entorno básico estación base – usuarios finales, que use [OMA](#) y [NOMA](#) y generar un conjunto de datos de entrenamiento, validación y prueba para la [ANN](#).
- Evaluar algoritmos de aprendizaje para una [ANN](#) y comparar el rendimiento de red móvil para cada uno de ellos.
- Evaluar la variación entre 1 y 3 capas ocultas en la [ANN](#) y comparar el rendimiento de red móvil para cada caso evaluado.
- Complementar con la evaluación de las funciones de activación de la red neuronal: lineal, sigmoidea y tangente hiperbólica, y posteriormente comparar el rendimiento de red móvil



para cada uno de ellos.



Capítulo 2

Marco Teórico

En este capítulo se presenta una revisión de los conceptos más relevantes y el sustento analítico que constituyen la base teórica del presente trabajo. Se describen las principales técnicas de conformación de haz (*beamforming*). Se analizan los modelos de canal de ondas milimétricas (*mm-Wave*). Se abordan las técnicas de acceso múltiple *OMA* y *NOMA*. Se describen las redes neuronales (*ANN*) y sus principales características de funciones activación y arquitecturas. Finalmente se repasan distintos modelos de entrenamiento de las *ANN*.



2.1. Técnicas de Acceso Multiple

En una red inalámbrica, los equipos de usuario comparten los recursos de la estación base. Estos recursos se denominan normalmente bloques de recursos (*Resource Blocks RB*) y permiten a los UEs recibir y transmitir información. Los RBs se distribuyen equitativamente entre los UEs utilizando diferentes métodos de acceso múltiple. Estos últimos se pueden clasificar principalmente en dos categorías: acceso múltiple ortogonal (OMA) y acceso múltiple no ortogonal (NOMA).

2.1.1. Acceso Múltiple Ortogonal (OMA)

Los métodos de acceso OMA son métodos de control de acceso múltiple sin contención. Los más utilizados son: el acceso múltiple por división de tiempo (TDMA), el acceso múltiple por división de frecuencia (FDMA), el acceso múltiple por división de código (CDMA) y el acceso múltiple por división de frecuencia ortogonal (OFDMA). La segunda generación de redes móviles utiliza una capa física basada en un híbrido TDMA/FDMA, la tercera generación se basa en CDMA y la cuarta generación en OFDMA. De acuerdo al enfoque (tiempo, frecuencia, o código), en todos los casos OMA se asigna un bloque de recursos exclusivo a cada UE [9].

2.1.1.1. Acceso Múltiple por División de Tiempo (TDMA)

En TDMA, el acceso de múltiples UEs a los RBs se lo realiza en base a la multiplexación por división de tiempo. El tiempo se divide en ranuras temporales de duración fija (*time slot*). Los UEs pueden transmitir o recibir datos únicamente en la ranura de tiempo que les ha sido asignada. Generalmente, los intervalos de tiempo están separados por pequeños intervalos de protección. En la Figura 2.1 se puede observar gráficamente la asignación exclusiva de las ranuras de tiempo a los distintos UEs. TDMA ha sido utilizado ampliamente por diversas tecnologías de comunicaciones móviles, especialmente la 2G.

Este método de acceso tiene una ventaja clara debido a su simpleza al momento de la implementación. Por otro lado, la principal desventaja está en el requisito de sincronización temporal rigurosa que debe existir entre el emisor y el receptor.

2.1.1.2. Acceso Múltiple por División de Frecuencia (FDMA)

En FDMA, a diferencia del método de acceso anterior, la división se realiza en el dominio de la frecuencia. Los recursos de radio se dividen en distintos canales de frecuencia lo suficientemente

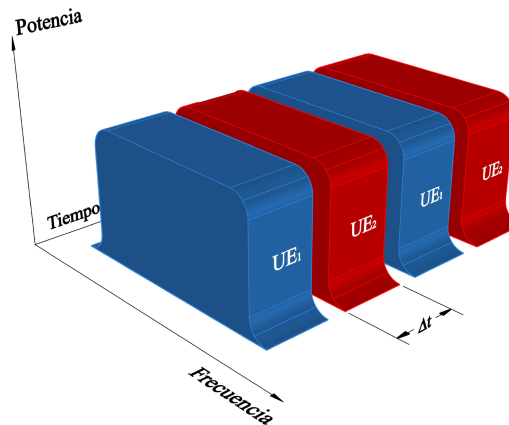


Figura 2.1: Método de acceso TDMA.

separados entre sí y comúnmente denominados subportadoras. Los canales son asignados a los UEs de acuerdo a la demanda. Con esta variación, cada UE puede realizar una transmisión ininterrumpida durante todo el intervalo de tiempo, pero en distintos canales de frecuencia [10]. En la Figura 2.2, se muestra la asignación de canales a los UEs.

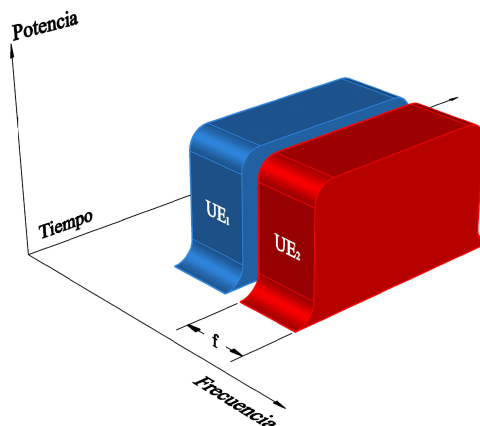


Figura 2.2: Método de acceso FDMA.

Al igual que TDMA, la ventaja de FDMA radica en la simpleza de la implementación y la robustez al multi-trayecto. Por otra parte, la escasa gestión para flujos de tráfico variable es una clara desventaja de este método de acceso múltiple [10].

2.1.1.3. Acceso Múltiple por División de Frecuencia Ortogonal (OFDMA)

OFDMA se asemeja a FDMA, con la distinción de que las subportadoras son ortogonales entre sí. Esto para minimizar las interferencias ocurridas entre los distintos UEs y para optimizar la

división del espectro [11]. En el dominio del tiempo, los símbolos OFDMA tienen una duración fija y se agrega un prefijo cíclico para reducir la interferencia intersimbólica. Cada símbolo OFDMA está compuesto por una cantidad específica de subportadoras y por tanto, OFDMA puede verse como un híbrido TDMA/FDMA.

En la Figura 2.3, se muestra la distribución de los RBs a los UEs. Como se mencionó anteriormente, tanto el eje temporal como el eje del espectro de frecuencias es dividido. Varias subportadoras son asignadas a cada UE de acuerdo a las condiciones del canal y los requerimientos del servicio. Esto se realiza con el objetivo de conseguir un uso eficiente de los recursos a nivel temporal y espectral [12]. Por tales motivos, OFDMA ha sido la base para la capa física de 4G.

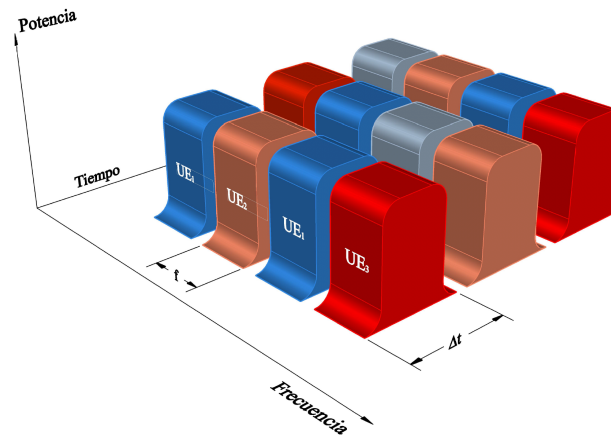


Figura 2.3: Método de acceso OFDMA.

2.1.2. Acceso Múltiple No Ortogonal (NOMA)

A diferencia de las técnicas de acceso múltiple ortogonales, NOMA asigna un bloque de recursos simultáneamente a más de un equipo de usuario, mejorando de esta manera la capacidad y eficiencia de la estación base. Para diferenciar entre los UEs que usan el mismo RB, NOMA puede utilizar información en el dominio de la potencia o en el dominio de código. Como es de esperarse, NOMA en el dominio de la potencia, asigna dependiendo de la condición del canal de cada usuario, un nivel de potencia diferente a cada UE [13]. Por su parte, NOMA en el dominio de código, utiliza secuencias de ensanchamiento (*spreading codes*) específicas para cada UE [14]. Es importante mencionar que en este trabajo solamente se considerará la opción de NOMA en el dominio de la potencia.

2.1.2.1. NOMA en el Dominio de la Potencia

En el dominio de la potencia la BS transmite una señal s a todos los UEs. La señal s , representada por la Ecuación 2.1, es la suma de todas las señales x_i de los UE que comparten un bloque de recursos. La potencia asignada a cada UE, representada con β_i , debe escogerse de una manera equitativa. Es decir, el UE con una peor condición de canal recibe un nivel de potencia mayor y así sucesivamente. A manera de ejemplo, en la Figura 2.4 se muestra una ilustración con dos usuarios con condiciones de canal \mathbf{h}_1 y \mathbf{h}_2 respectivamente, en este caso se asume que $\mathbf{h}_1 < \mathbf{h}_2$. Adicionalmente, P_s representa la potencia total. Es decir, la suma de potencias β_i y que generalmente se normaliza a 1 como se muestra en la Ecuación 2.2. Finalmente, \mathbf{w} es el vector de pesos utilizado en la formación de haces (*beamforming*) [7, 15].

$$s = \sum_{i=1}^K \mathbf{w} \sqrt{\beta_i P_s} x_i \quad (2.1)$$

$$\sum_{i=1}^K \beta_i = 1 \quad (2.2)$$

Como puede verse, un punto muy importante es la asignación adecuada de potencia. En [15, 16] se propone una estrategia, en la que se utiliza el inverso del canal de cada UE para asignar el nivel de potencia adecuado, como se muestra en la Ecuación 2.3.

$$\beta_i = \frac{1}{|\mathbf{h}_i^T \mathbf{w}|^2} \quad (2.3)$$

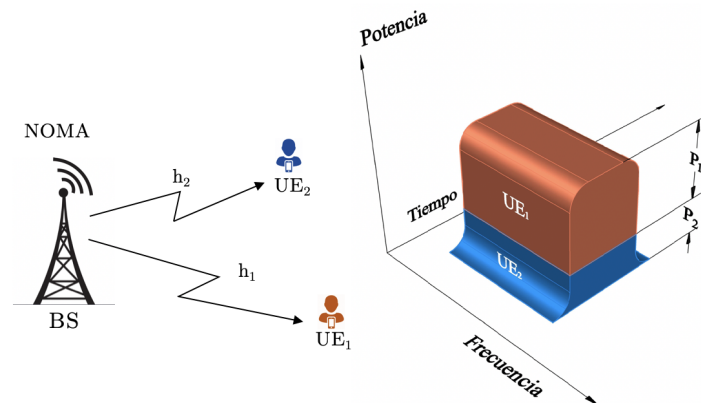


Figura 2.4: Ejemplo de asignación de potencia a cada UE

En el enlace descendente de **NOMA**, todos los **UEs** reciben s y emplean el mecanismo de cancelación de interferencia sucesiva (*Successive Interference Cancellation* **SIC**) para recuperar la señal individual de cada **UE**. En **SIC**, el **UE** con menor potencia decodifica todas las señales con mayor potencia y posteriormente resta el resultado de la señal total s para recuperar su propia señal.

Al utilizar **SIC**, el UE_k decodifica directamente su señal x_k y considera las otras señales como interferencia. El **UE** con el peor canal no necesitará hacer **SIC**, al contrario que el **UE** con el mejor canal que necesitará hacer **SIC**. Como se muestra en la Figura 2.5, el UE_1 primero necesita decodificar todas las señales más fuertes para finalmente restar el resultado de s y obtener la señal UE_1 [7, 17]. Desde el punto de vista computacional, esto podría representar una desventaja para el **UE** que necesita hacer **SIC**; sin embargo, como las capacidades de procesamiento de los nuevos dispositivos han aumentado considerablemente, esto ya no es un problema.

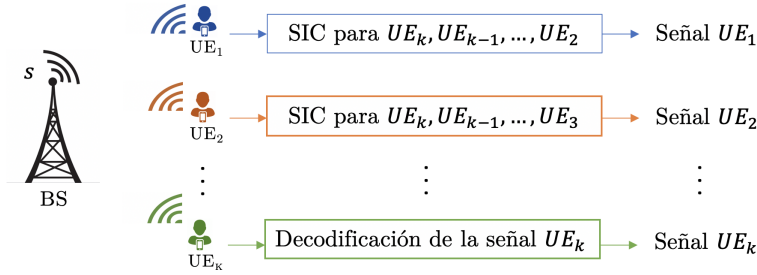


Figura 2.5: Decodificación en un enlace descendente de **NOMA**

La tasa total R_{sum} (*sum-rate*) en el enlace descendente **NOMA**, se logra definiendo la relación señal a ruido más interferencia **SINR** y la señal recibida del UE_K (Ecuación 2.4).

$$R_{sum} = \sum_{k=1}^K \log_2(1 + SINR_k) \quad (2.4)$$

Luego, combinando el *sum-rate* del UE_K , que no requiere **SIC**, y el *sum-rate* del resto de **UEs**, se obtiene la Ecuación 2.5.

$$R_{sum} = \sum_{k=1}^{K-1} \log_2 \left(1 + \frac{\beta_k}{\sum_{i=k+1}^K \beta_i + 1/\gamma |\mathbf{h}_k^T \mathbf{w}|^2} \right) + \log_2(1 + \beta_K \gamma |\mathbf{h}_K^T \mathbf{w}|^2) \quad (2.5)$$

donde, γ representa la **SNR** y $\mathbf{h}_K^T \mathbf{w}$ el n -ésimo canal **UE**.

Finalmente, debido a que en este trabajo solamente se utilizarán dos **UEs**, y considerando $|\mathbf{h}_1^T \mathbf{w}| \leq |\mathbf{h}_2^T \mathbf{w}|$, el *sum-rate* puede ser representado mediante la Ecuación 2.6.

$$R_{sum} = \log_2 \left(1 + \frac{\beta_1}{\beta_2 + 1/\gamma |\mathbf{h}_1^T \mathbf{w}|^2} \right) + \log_2 (1 + \beta_2 \gamma |\mathbf{h}_2^T \mathbf{w}|^2) \quad (2.6)$$

Como complemento, incluso si no está cubierto en este trabajo, es importante mencionar que **NOMA** no solo presenta una ventaja sobre **OMA** en el enlace descendente sino también en el enlace ascendente. El análisis matemático se diferencia en el hecho de que la **BS** recibe todas las señales ponderadas de los **UEs**. Luego, se aplica **SIC** como se muestra en [15].

2.2. Formación de Haces (Beamforming)

La formación de haz se puede definir como una técnica de procesamiento digital enfocada en mejorar la relación señal a ruido (*Signal-to-Noise Ratio* **SNR**) de un sistema de comunicación inalámbrico. La idea principal consiste en dirigir el haz de un conjunto de antenas en el transmisor en la dirección del receptor. A continuación se describen los tres posibles métodos de formación de haz: el *beamforming* analógico, el digital y el *beamforming* híbrido.

2.2.1. Beamforming Analógico

El *beamforming* analógico se genera introduciendo cambios de fase o de tiempo en la señal de cada uno de los elementos del arreglo de antenas en el transmisor. La Ecuación 2.7 muestra cómo se pueden generar dichos retrasos temporales [18].

$$\Delta t = \frac{d \cdot \sin(\theta)}{c} \quad (2.7)$$

donde, d es la distancia de separación entre los elementos del arreglo de antenas, usualmente $\lambda/2$, θ representa el ángulo requerido y c la constante de velocidad de la luz.

Los elementos de antena se pueden distribuir en cualquier configuración; sin embargo, el arreglo rectangular de antenas (*Uniform Rectangular Array* **URA**) es el más adecuado para las comunicaciones de próxima generación y, por lo tanto, se utilizará en este trabajo. Además de proporcionar el beneficio de atenuación de lóbulos laterales, esta antena en particular permite dirigir el haz en los planos de acimut y elevación [17–19].

En conclusión, al ser la técnica más estudiada y más utilizada por su fácil implementación es la seleccionada para este trabajo.

2.2.2. Beamforming Digital

El *beamforming* digital, en contraste con la arquitectura anterior, puede manejar un flujo de datos digitales por cada elemento que conforma la antena, posibilitando el envío de diferentes señales en paralelo. Esta característica proporciona varias ventajas como la generación de haz múltiple y una mayor flexibilidad. Pero, a pesar de las ventajas, esta arquitectura tiene algunas desventajas relacionadas principalmente con la complejidad de la implementación, el requisito de cadenas completas de transmisión por cada elemento de antena y el alto consumo de energía [19, 20].

2.2.3. Beamforming Híbrido

El *beamforming* híbrido es una combinación del *beamforming* analógico y digital. Esta técnica trata de mitigar las desventajas del *beamforming* digital, mediante la combinación con el *beamforming* analógico. El *beamforming* analógico añade los cambios de fase en cada flujo de datos digital, provocando una menor complejidad, pero reduciendo la cantidad de grados de libertad. Por tanto, la eficiencia esperada no alcanza al *beamforming* digital en el rango de frecuencias de ondas milimétricas [18, 20].

2.3. Canal de Ondas Milimétricas (mm-Wave)

En la literatura existen muchos modelos de canales inalámbricos, sin embargo, el modelo de canal de Rayleigh multi-trayecto [21] es el tradicionalmente utilizado para comunicaciones móviles de hasta 4G con frecuencias de operación menores a 5 GHz.

Debido a la saturación del espectro en las frecuencias de operación clásicas (< 5 GHz), las redes móviles de próxima generación se espera que operen en frecuencias de ondas milimétricas, por ejemplo, en torno a los 30 GHz. El modelo de canal que recoge más características en estas mm-Wave es el modelo LOS, y se muestra en la Ecuación 2.8 [22].

$$\mathbf{h}_i = \sqrt{M} \frac{\alpha_i a(\theta_i)}{\sqrt{1 + d_i^\phi}} \quad (2.8)$$

donde, M es el número de elementos del arreglo de antenas, α la ganancia compleja del enlace i (dado por una distribución Gaussiana con media 0 y varianza 1), a_1 el vector de dirección de *beamforming*, ϕ es el exponente de pérdida del enlace, y d es la distancia desde la BS hasta el



UE [22].

Como puede observarse, a diferencia de los modelos de canal clásicos, este modelo de canal utiliza únicamente la componente de línea de vista (*Line-of-Sight* LOS) de la señal recibida. Esto se justifica debido a la potencia dominante de la señal LOS sobre las componentes multi-trayecto sin línea de vista (*Non-Line-of-Sight* NLOS) [23]. Por esta razón, este modelo de canal LOS dado por la Ecuación 2.8 es el que se utiliza en este trabajo.

Adicionalmente, es importante mencionar que un requisito de NOMA es que los canales de los UE sean diferentes. Esto es necesario para evitar errores en el SIC al decodificar la señal de cada UE.

2.4. Red Neuronal Artificial (ANN)

Una red neuronal artificial (ANN) es un algoritmo de aprendizaje profundo inspirado en la estructura biológica de las neuronas cerebrales. El conjunto de neuronas biológicas conectadas entre sí logran comunicar y procesar información de todo el organismo. Cada neurona tiene la capacidad de modificar su configuración de acuerdo a la necesidad del organismo, con el objetivo de adaptarse a los distintos tipos de requerimiento, esto permite que la red aprenda y desarrolle memoria [24].

Una ANN utiliza un modelo matemático para asemejarse a la estructura de una neurona biológica. El modelo matemático procesa la información en la entrada y la relaciona con la salida deseada. La estructura consta de nodos o neuronas interconectadas, cada una procesa información y corrige los errores mediante un modelo matemático. Este modelo procesa la información en la entrada y la relaciona con las salidas. Este proceso en conjunto depende de las reglas de aprendizaje, las funciones de activación, y el algoritmo empleado para el entrenamiento [25].

2.4.1. Modelo computacional de una ANN

En el modelo computacional las neuronas se conectan entre sí por medio de capas, estas conexiones dependen de la arquitectura de la red. En la Figura 2.6 se muestra un esquema básico conocido como retro-propagación. Como puede observarse, cada neurona se conecta con todas las neuronas de la capa consecutiva. La primera capa está formada por las neuronas de entrada, en las cuales ingresan las características que conforman el patrón de entrada. Por otro lado, la última capa contiene las neuronas de la salida. Finalmente, entre la entrada y salida se encuentra la red interna, que puede estar formada por varias capas de neuronas denominadas capas

ocultas [26].

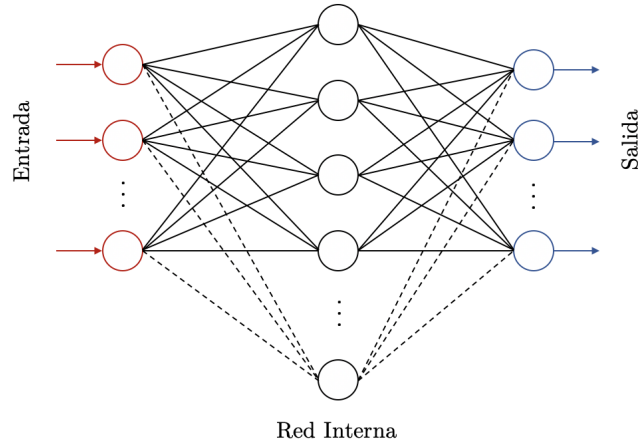


Figura 2.6: Ejemplo de un esquema básico de una ANN

En una ANN, el vector de entrada se propaga a través de toda la red, desde la entrada hasta generar la salida. La conexión entre neuronas modifica el valor de entrada, debido a los pesos que existen entre las conexiones de neuronas. Estos pesos son configurados iterativamente en la fase de aprendizaje para obtener una salida deseada.

Específicamente, el peso de la conexión w_{ij} se asocia a una neurona de la capa i conectada con otra neurona de la capa j . La red suma las entradas de una neurona y aplica una función de activación para generar la salida. Por tanto, la Ecuación 2.9 muestra cómo se podría definir la salida y_j de una neurona.

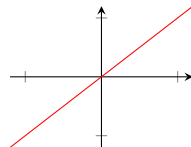
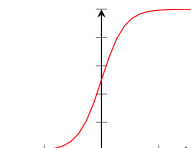
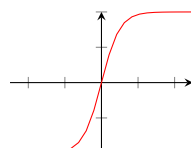
$$y_j = f(z_j) = f\left(\sum_{i=1}^n w_{ij}x_{ij} + b\right) \quad (2.9)$$

donde, x_{ij} representa las entradas, $f(z_j)$ es la función de activación aplicada a la salida z_j , y b es un umbral o valor de *bias* [27].

2.4.2. Función de Activación

Como se mencionó en la sección anterior, las señales de entrada son procesadas por una función de activación f . Las funciones de activación se dividen comúnmente en lineales y no lineales (ej. sigmoideal y tangente hiperbólica). Las funciones más utilizadas se muestran en la Tabla 2.1.

Tabla 2.1: Funciones de activación más utilizadas.

Función de Activación	Ecuación	Gráfico
Lineal	$f(z) = z \quad (2.10)$	
Sigmoidal	$f(z) = \frac{1}{1 + e^{-z}} \quad (2.11)$	
Tangente Hiperbólica	$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.12)$	

2.4.3. Algoritmos de entrenamiento

El algoritmo de entrenamiento sirve para modificar iterativamente los pesos de los enlaces. Esto con el fin de reducir la diferencia entre la salida de la red y la salida deseada (error de la ANN). Particularmente, para encontrar un mínimo global en la función, la modificación en los pesos w_{l+1} se realiza de manera proporcional a la derivada del error, como se define en la Ecuación 2.13. Los métodos de orden superior funcionan correctamente cuando la segunda derivada se calcula fácilmente.

$$w_{l+1} = w_l + \mathbf{H}_l^{-1} g_l \quad (2.13)$$

donde, w_l son los pesos actuales, \mathbf{H} es la matriz Hessiana de segundas derivadas [28] (Ecuación 2.14), y g_l es el gradiente de la función de error.

$$\mathbf{H} = \frac{\partial^2 f}{\partial w_p \partial w_q} \quad (2.14)$$

El cálculo de segundas derivadas tiene un costo computacional muy alto, debido a esto, se han desarrollado técnicas como el método de cuasi-Newton y el método de la secante que permiten aproximar las segundas derivadas [29]. Esto acelera el entrenamiento en cada época o iteración.

En este trabajo, la ANN tiene como objetivo clasificar a cada UE de cada grupo a un determinado método de acceso (OMA o NOMA). Este proceso tiene como fin el maximizar el rendimiento de la red móvil. Por lo tanto, este trabajo se basó en algoritmos de entrenamiento sugeridos por [30–34], que muestran opciones de entrenamiento con notable precisión y menor tiempo de entrenamiento para aplicaciones de clasificación. A continuación, se describen brevemente los algoritmos de entrenamiento de ANN evaluados en este trabajo.

2.4.3.1. Algoritmo de Broyden–Fletcher–Goldfarb–Shanno (BFGS)

BFGS es un algoritmo quasi-Newton que no calcula la matriz Hessiana. En su lugar, la aproxima utilizando actualizaciones obtenidas por las evaluaciones del gradiente [29], como se muestra en la Ecuación 2.15.

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{s_k s_k^T}{s_k^T r_k} - \frac{\mathbf{H}_k r_k r_k^T \mathbf{H}_k}{r_k^T \mathbf{H}_k r_k} \quad (2.15)$$

donde, se define $r_k = w_{k+1} - w_k$ y $s_k = \nabla f(w_{k+1}) - \nabla f(w_k)$.

Los métodos quasi-Newton tienden a converger en menos iteraciones que otros métodos basados en el gradiente, aunque requiere más tiempo de entrenamiento y más espacio de almacenamiento de memoria por iteración que otros métodos. Por esta razón, la complejidad del algoritmo BFGS es $O(N^2)$ [29].

2.4.3.2. Algoritmo Secante de un Paso (One-Step Secant OSS)

Comparado con BFGS, *One-Step Secant* OSS reduce el tiempo de entrenamiento y el almacenamiento en memoria requerido por iteración. Esto se logra al no guardar la matriz Hessiana completa. En cambio, se asume la matriz identidad como la matriz Hessiana de la iteración anterior. Como resultado los pesos y bias son ajustados en dirección del factor t_{k+1} [28]. En la Ecuación 2.16, los pesos y bias son representados por \mathbf{T} .

$$\mathbf{T}_{k+1} = \mathbf{T}_k + a \cdot t_{k+1} \quad (2.16)$$

donde, a es un escalar para limitar la búsqueda. La nueva dirección de búsqueda se calcula a partir del nuevo gradiente del error g_k , los pesos anteriores de la última iteración se definen como r_k y la diferencia de gradientes z_k . Dando como resultado la Ecuación 2.17, que muestra el factor de ajuste t_{k+1} .

$$t_{k+1} = -g_k + A_k r_k + B_k s_k \quad (2.17)$$

donde, los escalares A_k y B_k son definidos en las Ecuaciones 2.18 y 2.19 respectivamente.

$$A_k = \frac{s_k^T g_k}{r_k^T s_k} - B_k \left(1 + \frac{s_k^T s_k}{r_k^T s_k} \right) \quad (2.18)$$

$$B_k = \frac{r_k^T g_k}{r_k^T s_k} \quad (2.19)$$

Por todo lo mencionado anteriormente, la complejidad computacional de este algoritmo de entrenamiento se reduce a $O(N)$ [28].

2.4.3.3. Algoritmo de Levenberg-Marquardt (**L-M**)

L-M es una modificación del método de Newton. Este no calcula la matriz Hessiana, en su lugar realiza una aproximación mediante la matriz Jacobiana \mathbf{J} que contiene la derivada de los errores con respecto a pesos y bias. La matriz Jacobiana es menos compleja de calcular y utiliza menos recursos que una matriz Hessiana, como puede verse en la Ecuación 2.20 [28].

$$\mathbf{H} = \mathbf{J}^T \mathbf{J} \quad (2.20)$$

El gradiente g puede ser calculado usando el vector de errores de la red e y la matriz Jacobiana \mathbf{J} , como se muestra en la Ecuación 2.21.

$$g = \mathbf{J}^T e \quad (2.21)$$

En consecuencia, la actualización de pesos mostrada en la Ecuación 2.13 se convierte en la Ecuación 2.22.

$$w_{l+1} = w_l - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T e \quad (2.22)$$

donde, \mathbf{I} es la matriz identidad y μ es un escalar. Cuando μ tiende a cero, el algoritmo se comporta como el método de Newton y cuando μ tiende a un valor alto, el algoritmo se comporta como el método de descenso de gradiente [35]. En consecuencia, **L-M** requiere menos espacio de



almacenamiento y tiempo de entrenamiento.

En resumen, elegir el algoritmo de entrenamiento más adecuado es crucial para proporcionar a la ANN estabilidad en uso de memoria y tiempo de entrenamiento. Todo esto acorde a las necesidades de un entorno de selección de método de acceso más eficiente para redes móviles de próxima generación.



Capítulo 3

Trabajos relacionados

En este capítulo se destacan algunos trabajos relacionados con el acceso híbrido [OMA/NOMA](#) y el uso de [ANN](#) en aplicaciones relacionadas a las redes móviles de próxima generación. Esto se realiza con fin de analizar el estado del arte disponible y sustentar y diferenciar la contribución del presente trabajo.

Algunos de los desafíos de las redes de próxima generación son abordados utilizando técnicas de aprendizaje automático (*Machine Learning ML*), debido a la propiedad de alta extracción de características de datos sin procesar [36, 37]. Por ejemplo, los autores de [38] implementan una técnica de *ML* que trabaja en conjunto con una red de memoria a corto plazo (*Long Short-Term Memory Network LSTM*) para detectar las características cambiantes del canal. Otro problema también es abordado usando *LSTM* en [39]. Aquí, los autores estiman el canal y detectan múltiples usuarios para evitar la propagación de errores en el proceso de decodificación de secuencia del detector *SIC* en los receptores. Adicionalmente, en [40] también se estudia el problema relacionado con la detección y localización de usuarios. Pero en este caso, a diferencia de [39], los autores implementan una *ANN* entrenada con el algoritmo *L-M*.

Otro reto abordado con *ML* es la asignación óptima de potencia. La premisa aquí es que la potencia de un usuario debe asignarse de manera imparcial sin afectar al resto. En [41, 42], los autores proponen un algoritmo de agrupamiento de usuarios (clúster) que utiliza la técnica de *K-means*. Además, generan una política de asignación de potencia óptima mediante la explotación de la función de decodificación sucesiva y *SIC*. El objetivo es tener varios clústeres que no afecten al rendimiento de la red (interferencia entre clústeres). Tanto la estimación de canales, como la asignación de energía, son desafíos importantes que han sido abordados mediante el uso de técnicas de *ML*. Estos trabajos han servido de motivación para el uso de técnicas de *ML* aplicadas a solucionar los desafíos de un mecanismo de control de acceso *NOMA*.

Otro problema esencial de los sistemas *NOMA* radica en escenarios en los que las condiciones del *UE* son heterogéneas (ej. estado del canal y *SNR*). Por tanto, es muy importante estudiar la combinación de *NOMA* con otros métodos de acceso [2]. Los autores de [6], introducen el concepto *SoDeMa* con el objetivo de presentar las pautas que debe tener un sistema para seleccionar el mejor método de acceso en cada *UE*. Considerando además que el sistema tenga un rendimiento óptimo. La idea principal es tener una adaptación flexible según los requisitos del usuario, las condiciones del entorno de la red y el tipo de servicio o aplicación. En consecuencia, surgen los esquemas cooperativos *OMA* y *NOMA* como una solución clave al desafío antes mencionado.

En este contexto, los autores en [43], proponen un esquema cooperativo con el objetivo de mejorar el desempeño del *UE* más lejano a la *BS*, sin descuidar el desempeño del *UE* cercano. Este rendimiento no debe estar por debajo del obtenido por un sistema *OMA*. Por otro lado, los trabajos [44–46] presentan esquemas híbridos *OMA/NOMA* asistidos por un búfer para esquemas de transmisión de enlace descendente y ascendente. Naturalmente, se basan en el estado del búfer y en la información instantánea del estado del canal (*Channel Status Information CSI*). Cuando el búfer no puede cumplir con los requisitos, cambia el método de acceso dando prioridad a *NOMA*.



Como se puede notar, el enfoque de aprendizaje automático ha mostrado resultados notables aplicados a diferentes desafíos de redes móviles. Bajo estas premisas, este trabajo propone un novedoso sistema de gestión de acceso OMA/NOMA basado en ML. La propuesta utiliza una ANN para decidir la mejor combinación entre OMA y NOMA como método de control de acceso para los UEs. Adicionalmente, como parte de este trabajo, se evalúa la ANN con tres algoritmos de entrenamiento: L-M, BFGS y OSS, junto con diferentes arquitecturas y funciones de activación. El objetivo es determinar la mejor configuración de ANN que brinde una reacción rápida junto con una baja complejidad computacional. En resumen, la presente propuesta busca un uso eficiente del espectro de radio y un alto equilibrio de rendimiento de un sistema móvil de próxima generación.



Capítulo 4

Metodología

En este capítulo se presenta la metodología utilizada para el desarrollo de este trabajo.

En este trabajo, se propone un sistema el que la estación base gestiona automáticamente el método de acceso para cada equipo de usuario entre un acceso ortogonal o no ortogonal. Esta selección entre OMA o NOMA se toma en la BS utilizando una red neuronal artificial adaptada para esta aplicación específica. Se espera entonces, que la BS seleccione los casos que presenten el mejor rendimiento de la red. La coexistencia armónica entre los dos métodos de control de acceso tiene como objetivo generar un uso más eficiente de los bloques de recurso radio. Estos factores contribuyen a la mejora de la calidad del servicio y la calidad de la experiencia percibida por los usuarios finales.

Para describir a detalle la metodología del sistema híbrido propuesto en este trabajo, se ha dividido el proceso completo en los siguientes pasos:

1. Especificación del sistema híbrido OMA/NOMA
2. Generación del conjunto de datos
3. Sistema de gestión mediante una ANN
4. Criterios de evaluación del desempeño

4.1. Especificación del Sistema Híbrido OMA/NOMA

Este trabajo toma como principio y motivación los resultados obtenidos en [15] para sistemas NOMA. En dicho trabajo previo, se demostró que el mayor rendimiento de NOMA se logra en un entorno de dos equipos de usuario por cada haz. Además, se indica que el haz más directivo se genera mediante un arreglo rectangular uniforme (URA), específicamente un arreglo de 25 elementos.

Por tanto, en esta propuesta de coexistencia OMA/NOMA, a más de los dos UEs con acceso NOMA se considera un entorno con al menos un tercer UE con acceso OMA. En este contexto, en este trabajo se consideró un entorno definido y controlado conformado por: una BS que dirige dos haces hacia los UEs utilizando *beamforming* analógico, dos UEs que compartirán un haz con acceso NOMA, y un tercer UE que utiliza un solo haz con acceso OMA. Además, se asume que la BS tiene una cobertura de 500 m [7] y que opera a una frecuencia de 32 GHz perteneciente al rango de ondas milimétricas. En la Figura 4.1, se muestra el entorno utilizado, y se compara con un sistema OMA tradicional. Puede verse además la utilización y asignación de los bloques de recurso en cada caso.

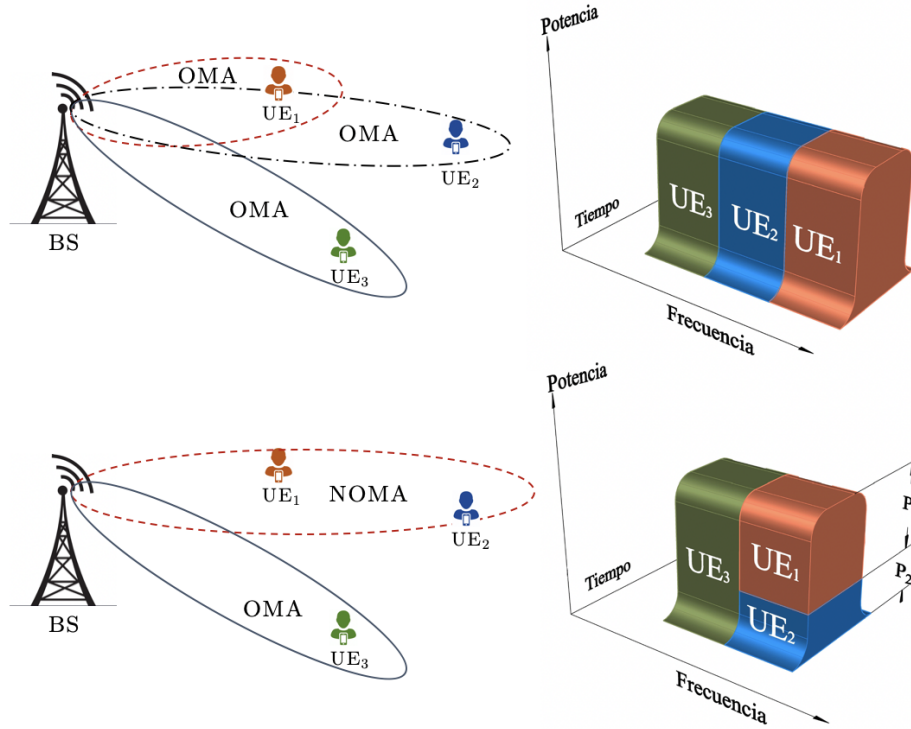


Figura 4.1: Comparación entre un sistema con acceso OMA y un sistema con acceso híbrido OMA/NOMA.

4.2. Generación del Conjunto de Datos

Inicialmente, cada UE se distribuyó aleatoriamente dentro de un radio de 500 m. Cada simulación se repitió 5 000 veces por cada valor de SNR definido entre 0 dB y 50 dB y dividido en intervalos de 2 dB. Esto produce un total de 125 000 posiciones de cada UE. Cada posición de UE proporciona datos del ángulo de incidencia (azimuth y elevación). Para la simulación y la obtención de resultados de este escenario se utilizó el software Matlab® (2019b), debido a sus altas prestaciones para cálculos matemáticos y librerías dedicadas a las comunicaciones móviles. Esta primera parte se consolida en un script disponible en el Apéndice B.1.

En base a lo anterior, el sistema es capaz de establecer el canal inalámbrico LOS [42], mediante la Ecuación 2.8. Además, se realiza la conformación del haz mediante el ajuste de fase de las ondas milimétricas (mm-Wave) presentes en cada elemento del arreglo de antenas. Para realizar este proceso se crearon funciones independiente como las que se muestran a continuación:

- Canal
- Formación de Haces (*beamforming*)

```
function ch=channel(N,UE,alpha1,alpha2,arrayURA,freq,UE_angles_N)
    %Steering vector:
    steervec = phased.SteeringVector('SensorArray',arrayURA);

    a_theta_1 = steervec(freq,UE_angles_N(:,1));
    a_theta_2 = steervec(freq,UE_angles_N(:,2));

    %Channel:
    alpha1=(normrnd(0,1)+1i*normrnd(0,1))/sqrt(1+((0.5-0.01)*rand(1)+0.01)^2);
    alpha2=(normrnd(0,1)+1i*normrnd(0,1))/sqrt(1+((0.5-0.01)*rand(1)+0.01)^2);
    ch = zeros(length(a_theta_1),UE);
    ch(:,1)=sqrt(N)*alpha1*a_theta_1;
    ch(:,2)=sqrt(N)*alpha2*a_theta_2;
end
```

```
function w = phase_shift_bf(arrayURA,inputAngle)
    global freq, lambda;
    N=5;
    arrayURA = phased.URA('Size',[N N],'ElementSpacing',[0.5*lambda 0.5*lambda]);
    arrayURA.Element.FrequencyRange = [30e9 34e9];

    % Phase Shift Beamformer
    psbeamformer = phased.PhaseShiftBeamformer('SensorArray',...
        arrayURA,'OperatingFrequency',freq,'Direction',...
        inputAngle,'WeightsOutputPort', true);
    [~,w] = psbeamformer(collectPlaneWave(arrayURA,1,inputAngle,freq));
end
```

De la misma manera, se utiliza el inverso del canal de cada UE con el objetivo de realizar una asignación justa de la potencia en NOMA [16]. Con base en la Ecuación 2.3, se elaboró la función de asignación de potencia.

```
function beta=powr(ch)
    P_UE1= 1/(snr*abs(ch(:,1))*w)^2);
    P_UE2= 1/(snr*abs(ch(:,2))*w)^2);
    beta(1)=P_UE1/(P_UE1+P_UE2)-0.2;
    beta(2)=1-beta(1);
end
```

Finalmente, se calcula la tasa de transmisión total (*sum-rate*) del sistema para el enlace descendente mediante la Ecuación 2.6, para las todas las combinaciones posibles de tres UEs mostradas en la Tabla 4.1.

**Tabla 4.1:** Combinaciones posibles de UEs

Combinación	UE_1	UE_2	UE_3
1	NOMA	NOMA	OMA
2	NOMA	OMA	NOMA
3	OMA	NOMA	NOMA

```

function sum=sum_rate_NOMA(N_ant,w,a_theta_1,a_theta_2,snr,alpha1,alpha2)
global UE_NOMA
H=channel(N_ant,a_theta_1,a_theta_2,UE_NOMA,alpha1,alpha2);
GUE_1=snr*abs(H(:,1)'*w)^2;
GUE_2=snr*abs(H(:,2)'*w)^2;
Beta=powr(GUE_1,GUE_2);
sum=log2(1+((Beta(1)*GUE_1)/(GUE_1*Beta(2)+1)))+log2(1+Beta(2)*GUE_2);
end

function sum=sum_rate_OMA(O_ant,w,a_theta_1,snr,alpha1)
global UE_OMA
H=channel(O_ant,a_theta_1,0,UE_OMA,alpha1,0);
sum=log2(1+(snr*abs(H(:,1)'*w)^2));
end

SumRate= SR_URA_NOMA + SR_URA_OMA

```

El conjunto de datos se obtiene de una selección de la combinación **OMA/NOMA** con el valor más alto de *sum-rate*. Esto sirve para generar el conjunto de valores que se utilizarán para entrenar, validar y probar a la **ANN**. Tanto la entrada de datos como la selección de los mismos se muestra en el script del Apéndice B.2. En la Tabla 4.2 se presenta el ejemplo de un conjunto de datos resultante para un caso particular, separados por parámetros de entrada y salida.

Tabla 4.2: Ejemplo numérico de datos a utilizar en el entrenamiento y/o validación de la **ANN**.

UE	Datos de Entrada				Datos de Salida
	Azimuth [deg] [-90,90]	Elevación [deg] [-90,90]	Canal [dB] $\mathcal{CN}(0, 1)$	SNR [dB] [0,50]	Método de Acceso
1	85.51	-10.51	$1,92 + 0,11i$	26	NOMA
2	36.16	51.98	$-1,07 - 1,43i$	26	OMA
3	-15.89	-64.64	$0,71 - 1,50i$	26	NOMA

4.3. Sistema de Gestión ANN

La tarea de la ANN es elegir una de las tres posibilidades (Tabla 4.1) de proporcionar acceso híbrido para maximizar el *sum-rate* del sistema. Por lo tanto, el objetivo en este punto es encontrar la mejor topología de red neuronal, junto con el algoritmo de entrenamiento más eficiente, y la función de activación adecuada.

El conjunto de datos se obtiene como se describe en la Sección 4.2. Estos se dividieron aleatoriamente en tres categorías:

- 70 % de los datos se utilizaron para el entrenamiento,
- 15 % para la validación, y el
- 15 % restante para las pruebas y evaluación.

Posteriormente, el conjunto de datos se pasa por una etapa habitual de pre-procesamiento, que básicamente consiste en normalizar los datos y eliminar duplicados. La normalización tiene por objetivo dar a cada característica la misma influencia en la salidas deseadas. La eliminación de datos duplicados mejora el entrenamiento con la reducción de la cantidad de datos, y porque solo se quedan los elementos más representativos.

En este trabajo, se contempla un clasificador no paramétrico conocido como red *feedforward* multicapa. Se ha considerado como la mejor opción para aplicarlo en entornos como el de la presente propuesta [33]. Para llevar a cabo este sistema de gestión ANN se utilizó el *ANN Toolbox* de Matlab® (2019b). Este software facilita la construcción de distintas arquitecturas, y la aplicación de distintos algoritmos de entrenamiento y funciones de activación [47]. Todas las ejecuciones se llevaron a cabo en un procesador Intel® Core™ i7-6700HQ y memoria RAM de 16 GB.

En la red *feedforward* se evaluaron de 1 a 3 capas ocultas y se seleccionó el número de neuronas por capa oculta (HL). Esto se realizó mediante un análisis previo utilizando una menor cantidad de datos (50 % del conjunto total de datos) con el objetivo de seleccionar las arquitecturas más significativas para analizarlas (Apéndice A). Finalmente, se optó por 9 arquitecturas, 3 por cada número de HL, como se indica en la Tabla 4.3.

Además, se evaluaron dos distintas función de activación en las capas ocultas; primero una función sigmoideal estándar (Ecuación 2.11) y luego una función tangente hiperbólica (Ecuación 2.12). Mientras que en la capa de salida, se utilizó una única función lineal (Ecuación 2.10). Estas funciones de activación son las configuraciones más utilizadas y recomendadas, debido a que obtienen mejores resultados para clasificadores [30, 34, 48].

**Tabla 4.3:** Arquitecturas ANN seleccionadas

Número de HL	Neuronas		
	1. ^a HL	2. ^a HL	3. ^a HL
1	20	-	-
	35	-	-
	50	-	-
2	25	10	-
	35	15	-
	40	20	-
3	40	20	10
	45	25	15
	50	30	20

Así mismo, se seleccionaron tres algoritmos de entrenamiento para la ANN: BFGS, OSS y L-M; debido a que ofrecen mejores resultados en clasificación que otros métodos [30–34], los mismos fueron detallados en la Sección 2.4.3.

Es también importante mencionar que todas las ANNs tienen un criterio de detenimiento que se alcanza cuando el número de iteraciones supera las 1000, el error cuadrático medio alcanza un valor mínimo de $1e^{-10}$, o el error cuadrático medio aumenta 6 veces consecutivas. Por último, cada algoritmo de entrenamiento se ejecutó 25 veces para cada arquitectura definida (Tabla 4.3) y 25 veces más por cada función de activación de las HL. Esto se realizó únicamente para las dos mejores arquitecturas con un algoritmo de entrenamiento, como se detalla en el Capítulo 5. En consecuencia, en total se entrenaron y analizaron 725 ANNs. Lo explicado se implementa principalmente a través del código mostrado a continuación.

```
%Data:
in = Data_In';
out = Data_Out';
repeat = 25;
%Create Network:
hiddenLayerSize = [HL1,HL2,HL3];
Train_Func=string({'trainlm','trainbfg','trainoss'});
TransferFcn1 = 'tansig'; %or
TransferFcn2 = 'logsig';
HLi = [2,3,4];
for i=1:repeat
    net = feedforwardnet(hiddenLayerSize);
    %Setup Division of Data for Training, Validation, Testing:
    net.divideFcn = 'dividerand'; % Divide data randomly
    net.divideMode = 'sample'; % Divide up every sample
    net.divideParam.trainRatio = 70/100;
    net.divideParam.valRatio = 15/100;
```



```
net.divideParam.testRatio = 15/100;
net.layers{HLi}.transferFcn = TransferFcn1;
%Train the Network:
for j=1:length(Train_Func)
    %Choose Input and Output Pre-Processing Functions:
    net.input.processFcns = {'removeconstantrows','mapminmax'};
        % mapminmax: normalizaci3n entre -1 y 1
        % removeconstantrows: Eliminar filas de matriz con valores repetidos.
    net.trainFcn = Train_Func(j);
    %Choose Perform Function:
    net.performFcn = 'mse';
    rng(seed);
    net.trainParam.showWindow=0;
    [net,tr] = train(net,in,out,'useParallel','yes','useGPU','only');
    nntraintool('close');
    y = net(in);
    %Reset(weights,bias) ...
end
end
```

4.4. Criterios de Evaluaci3n del Desempeño

En este trabajo se evalu3 el desempeño tanto de la ANN como del sistema híbrido OMA/NOMA en su conjunto. La ANN se evalu3 a nivel de arquitectura, algoritmo de entrenamiento y funci3n de activaci3n. Para cada uno de estos se utiliz3 la métrica de tiempo de entrenamiento, precisi3n y número de iteraciones. La precisi3n se define como la divisi3n entre los verdaderos positivos y la suma de verdaderos positivos con falsos positivos. Posteriormente, los resultados obtenidos de la ANN son aplicados al entorno móvil de acceso híbrido OMA/NOMA. Y este sistema híbrido OMA/NOMA gestionado por la ANN fue comparado con un sistema de selecci3n aleatoria de método de acceso, utilizando como criterio de evaluaci3n el *sum-rate* total del sistema en cada caso.

El tiempo de entrenamiento y los resultados de precisi3n de cada caso evaluado de la ANN se analizaron estadísticamente utilizando la mediana y la desviaci3n estándar. La mediana se utiliz3 debido a que existen valores atípicos que pueden influir mucho en el caso de utilizar la media. Adicionalmente, para evaluar el número de iteraciones de cada algoritmo de entrenamiento y el efecto del cambio de funci3n de activaci3n, se realiz3 un diagrama de caja (*box-plot*).

Finalmente, se selecciona la ANN que ofrece el menor tiempo de entrenamiento y la ANN con mayor precisi3n para tener opciones de resultados que sobresalgan en distintas características. Estas se analizaron y compararon mediante el gráfico de *sum-rate* del sistema en funci3n de la



UNIVERSIDAD DE CUENCA

SNR.



Capítulo 5

Resultados

En este capítulo se presentan los resultados de la evaluación del desempeño de la red neuronal artificial con distintas arquitecturas, algoritmos de entrenamiento y funciones de activación. Además, se evalúa el rendimiento del sistema híbrido [OMA/NOMA](#) y se compara con un sistema de selección aleatoria de método de acceso para cada [UE](#).



5.1. Análisis de Desempeño de la ANN

5.1.1. Análisis del Tiempo de Entrenamiento

Como se mencionó anteriormente, las arquitecturas ANN se entrenaron utilizando los algoritmos L-M, BFGS y OSS. El proceso de entrenamiento arrojó los resultados que se muestran en la Tabla 5.1 respecto al tiempo de entrenamiento. En esta se indica la mediana M_e y la desviación estándar σ para cada arquitectura de neuronas y para cada opción de capas ocultas. Los algoritmos L-M y BFGS muestran los tiempos de entrenamiento más cortos para el caso de una ANN con 20 neuronas en 1 HL (28 s y 29 s, respectivamente). Sin embargo, BFGS muestra una ligera ventaja en términos de variabilidad debido a una desviación estándar σ de 0.01 s frente a los 0.2 s de L-M. A partir de esta arquitectura en adelante, el algoritmo OSS es el que presenta el menor tiempo de entrenamiento para todas las arquitecturas analizadas (mínimo de 44 s y un máximo de 2 minutos con 10 segundos).

Por otro lado, se alcanzaron aproximadamente 71 minutos de tiempo de entrenamiento con el algoritmo BFGS para la ANN de 50-30-20 neuronas en 3 HL. Esto indica que este algoritmo de entrenamiento presenta un aumento considerable de tiempo de entrenamiento conforme se incrementa la cantidad de neuronas y de capas ocultas. Caso similar sucede con el algoritmo L-M que indica un incremento importante del tiempo de entrenamiento bajo estas condiciones.

En resumen, la complejidad computacional de los algoritmos se ve refleja en el tiempo de entrenamiento. OSS presenta una menor complejidad computacional, y como se menciona en [28], BFGS y L-M muestran una complejidad computacional más alta.

Tabla 5.1: Resultados del tiempo de entrenamiento de la ANN.

Capas Ocultas	Neuronas por Capa Oculta	BFGS		OSS		L-M	
		M_e [mm:ss]	σ [s]	M_e [mm:ss]	σ [s]	M_e [mm:ss]	σ [s]
1	20	00:29	0.01	00:44	0.01	00:28	0.20
	35	01:05	0.01	00:56	0.01	01:55	0.50
	50	02:14	0.02	01:29	0.02	06:55	0.90
2	25-10	01:41	0.02	01:19	0.02	03:41	0.20
	35-15	04:12	0.15	01:29	0.02	11:50	1.61
	40-20	09:36	0.54	01:27	0.02	28:53	3.29
3	40-20-10	14:20	0.55	01:40	0.02	21:29	1.86
	45-25-15	33:38	4.11	01:50	0.03	26:46	2.11
	50-30-20	70:58	16.08	02:10	0.03	37:16	2.21

5.1.2. Análisis de la Precisión del Entrenamiento

Para complementar el análisis del tiempo de entrenamiento, es necesario estudiar la precisión de cada arquitectura ANN. La Tabla 5.2 presenta la precisión, mediante la mediana M_e y la desviación estándar σ , esto para cada arquitectura de neuronas y opción de HL. En esta tabla se puede observar que el algoritmo L-M obtiene el mejor resultado general. L-M muestra una efectividad máxima de 76,97 % y una σ de 0.93 para la ANN con 50-30-20 neuronas en 3 capas ocultas.

Por otro lado, BFGS exhibe un incremento sostenido y continuo de la precisión hasta la ANN con 40-20-10 en 3 HL, más allá de esta red se evidenció una disminución de precisión. Particularmente, BFGS indicó una disminución en la ANN con 25-10 neuronas en 2 HL de 73,97 %, en comparación con las redes de 1 HL (35 y 50 neuronas). Sin embargo, este valor se compensa con una menor variabilidad con una σ de 1,82 %, que abarca las redes ya mencionadas de 1 HL.

Un caso particular el de OSS debido a que evidenció la peor precisión (72,45 %) y una máxima de 73,89 % (sin superar a L-M), esto a pesar de su velocidad en el tiempo de entrenamiento (Tabla 5.1). Con esta red no se logró mantener un incremento de precisión conforme se aumenta la cantidad de neuronas y de capas ocultas.

Tabla 5.2: Resultados de la precisión de la ANN.

Capas Ocultas	Neuronas por Capa Oculta	BFGS		OSS		L-M	
		M_e [mm:ss]	σ [s]	M_e [mm:ss]	σ [s]	M_e [mm:ss]	σ [s]
1	20	72.61	0.55	72.45	0.48	73.17	0.15
	35	74.15	0.29	73.26	0.80	74.32	0.18
	50	74.38	0.15	73.41	0.79	74.76	0.19
2	25-10	73.97	1.82	72.94	1.15	74.83	1.13
	35-15	74.36	1.21	73.28	0.67	75.57	0.44
	40-20	74.69	1.13	73.25	0.73	76.11	0.44
3	40-20-10	75.16	1.73	73.89	0.76	76.17	1.74
	45-25-15	75.09	1.38	73.04	0.67	76.95	1.56
	50-30-20	74.42	1.39	73.48	0.47	76.97	0.93

A partir de este punto, este trabajo se centró en analizar la ANN que presenta un menor tiempo de entrenamiento y aquella que tiene mayor precisión. Por lo tanto, para el análisis se selecciona la ANN con 20 neuronas en 1 HL y la ANN con 50-30-20 neuronas en 3 HL, respectivamente.

5.1.3. Análisis de la Cantidad de Iteraciones del Entrenamiento

En la Figura 5.1, se puede observar el número de iteraciones necesarias para cada algoritmo de entrenamiento mediante un diagrama de caja (*box-plot*). Se muestra de color rojo la ANN con 1 HL (arquitecturas que presentaron menor tiempo de entrenamiento) y de color azul la ANN con 3 HL (arquitecturas que presentaron mayor precisión). En el caso del algoritmo L-M se observa que con el incremento en el número de neuronas y de HL, el algoritmo necesita menos iteraciones para alcanzar la mayor precisión (una mediana de 35 iteraciones para 3 HL). Sin embargo, el costo computacional es más alto en cada iteración (tiempo de entrenamiento, Tabla 5.1).

Para el algoritmo BFGS sucede lo contrario, este requiere un mayor número de iteraciones a medida que incrementan tanto la cantidad de neuronas como el número de HL, esto también significa un incremento en la precisión.

Por otro lado, el algoritmo OSS con 3 HL presenta un número de iteraciones concentrados dentro del primer y tercer cuartil de la ANN con 1 HL. Es decir, a medida que aumentan las neuronas y las capas ocultas, las iteraciones se concentran aproximadamente en la media.

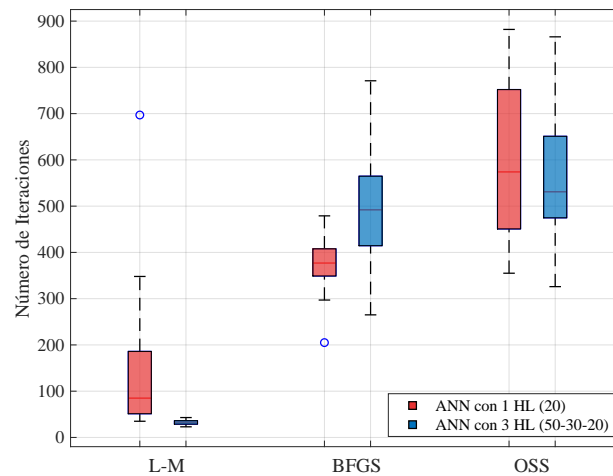


Figura 5.1: Diagrama de caja del número de iteraciones de ANN para 1 HL y para 3 HL.

5.1.4. Análisis de la Función de Activación

En esta sección se analiza el efecto que tiene el cambio de la función de activación en las neuronas de las capas ocultas. Las funciones de activación utilizadas son: sigmoide (Ecuación 2.11) y tangente hiperbólica (Ecuación 2.12).

En la Figura 5.2, se muestran los resultados de la arquitectura de ANN que obtuvo el menor tiempo de entrenamiento (20 neuronas en 1 HL). En color rojo se presentan los resultados para la función de activación tangente hiperbólica y en color azul los resultados para la función de activación sigmoide. La precisión se observa en la Figura 5.2a, esta no evidencia mayores cambios tanto en L-M y BFGS debido a que los resultados se encuentran dentro de los límites máximos y mínimos. Pero, OSS exhibe una ligera mejora de 0.8 % en la media, por parte de la función de activación tangente hiperbólica. Por otro lado, el tiempo de entrenamiento se muestra en la Figura 5.2b, esta indica pequeños cambios de alrededor de 10 s en la media, para todos los algoritmos de entrenamiento.

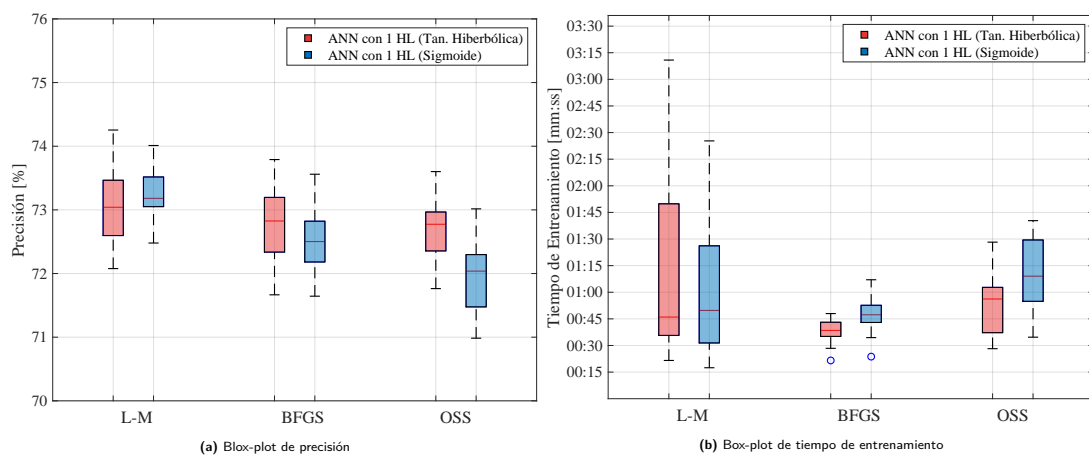


Figura 5.2: Resultados del cambio de función de activación (sigmoide y tangente hiperbólica) para la ANN de 20 neuronas en 1 HL.

De la misma manera, en la Figura 5.3 se presentan los resultados de la arquitectura de ANN que obtuvo la mayor precisión (50-30-20 neuronas en 3 HL). La precisión se presenta en la Figura 5.3a. Para el algoritmo L-M se puede observar que se tiene el rango inter-cuartil semejante y una media alrededor de 77 % para ambas funciones de activación. No obstante, BFGS y OSS exhiben ligeras mejoras del rango inter-cuartil con la función de activación tangente hiperbólica de alrededor del 1 %. Otro punto esencial es el tiempo de entrenamiento mostrado en la Figura 5.3b, esta presenta resultados muy semejantes de media en los 3 algoritmos de entrenamiento. Por otro lado, BFGS amplió su rango inter-cuartil con la función de activación sigmoide, indicando que existe una mayor variación en los tiempo de entrenamiento, esto no es conveniente debido que en esta arquitectura de ANN los tiempos de entrenamiento son altos (por encima de los 30 minutos) y es preferible evitar esto.

Estos resultados nos motivan a utilizar la función de activación tangente hiperbólica, debido a que ha presentado mejores características tanto en precisión como en tiempo de entrenamiento para la mayoría de los algoritmos de entrenamiento. De ahora en adelante se utilizará la ANN con esta función de activación para analizar el desempeño de la red móvil mediante el sistema

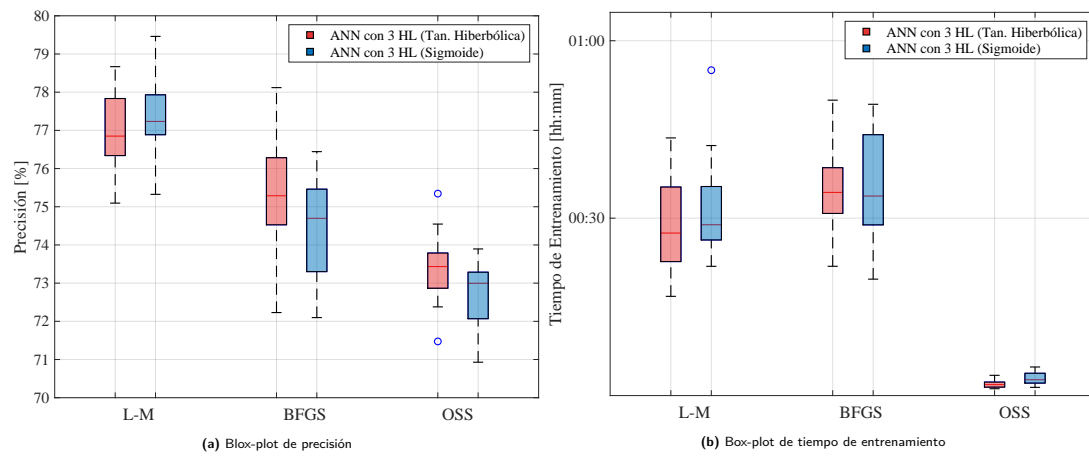


Figura 5.3: Resultados del cambio de función de activación (sigmoide y tangente hiperbólica) para la ANN de 50-30-20 neuronas en 3 HL.

de gestión de control de acceso híbrido propuesto.

5.2. Análisis del Desempeño del Sistema Híbrido OMA/NOMA

Para analizar el desempeño de la red móvil con el sistema híbrido OMA/NOMA se utilizan las salidas obtenidas de la red neuronal que indicará el método de acceso que debe utilizar cada UE. Luego, se calcula el *sum-rate* del sistema utilizando la Ecuación 2.6 y se representan gráficamente los resultados.

5.2.1. Análisis del Desempeño del Sistema Híbrido OMA/NOMA administrado por la ANN con el Menor Tiempo de Entrenamiento.

La Figura 5.4 muestra el *sum-rate* en función de la SNR para un sistema híbrido OMA/NOMA administrado por una ANN con 20 neuronas en una capa oculta. La ANN entrenada por el algoritmo L-M se muestra en color rojo, el algoritmo BFGS en color azul, el algoritmo OSS en color verde y el acceso híbrido OMA/NOMA sin gestión (es decir, la elección aleatoria) en color magenta. Al interior de la Figura 5.4, se puede ver una aproximación al intervalo de SNR entre 36 dB y 46 dB. En esta se observa, que las ANNs con los tres algoritmos de entrenamiento muestran resultados similares, con ligeros cambios en el *sum-rate*. Para complementar el análisis, la Tabla 5.3 resume el porcentaje promedio de aumento del rendimiento de cada ANN. En este caso, los tres algoritmos están cerca de un 20 % de mejora con respecto al sistema híbrido OMA/NOMA sin gestión. Con esto se cumple el principal objetivo de la presente propuesta que es el de optimizar la selección del método de acceso de tal forma que permita un uso más eficiente de los recursos radio, lo cual es expresado en este caso en términos de la tasa total del sistema (*sum-rate*).

El mejor algoritmo de entrenamiento fue BFGS superando al resto con aproximadamente un 0,095 %. Así, con base en estos resultados apoyados por el desempeño de la ANN, tanto BFGS como L-M se mostraron como los más favorables debido a su bajo tiempo de entrenamiento.

5.2.2. Análisis del Desempeño del Sistema Híbrido OMA/NOMA administrado por la ANN con Mayor Precisión.

De la misma manera, la Figura 5.5 muestra el *sum-rate* en función de la SNR para el sistema híbrido OMA/NOMA gestionado por una ANN con 50-30-20 neuronas en 3 HL. L-M presenta un claro incremento de alrededor de 0.2 bps/Hz por cada SNR con respecto a BFGS y OSS. En este

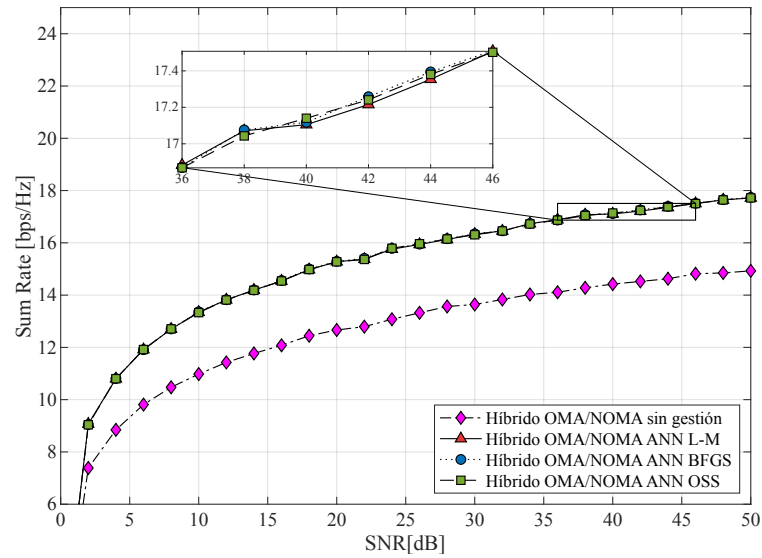


Figura 5.4: Rendimiento del sistema híbrido OMA/NOMA gestionado por una ANN con 20 neuronas en 1 HL.

caso, se muestra en la Tabla 5.3 que L-M obtuvo una mejora del 22,802 % con respecto al sistema híbrido OMA/NOMA sin gestión, mientras que BFGS obtiene una mejora del 21,152 %. Se observa además que L-M es el algoritmo que tiene un rendimiento eficiente para la clasificación de UE por método de acceso y que maximiza el sum-rate de la red móvil, aunque esto a costa de un tiempo de entrenamiento más alto (alrededor de 37 minutos).

Tabla 5.3: Aumento promedio del rendimiento del sistema híbrido OMA/NOMA.

Algoritmo de Entrenamiento	ANN con 1 HL (20)	ANN con 3 HL (50-30-20)
BFGS	20.089 %	21.152 %
OSS	19.994 %	20.474 %
L-M	19.995 %	22.802 %

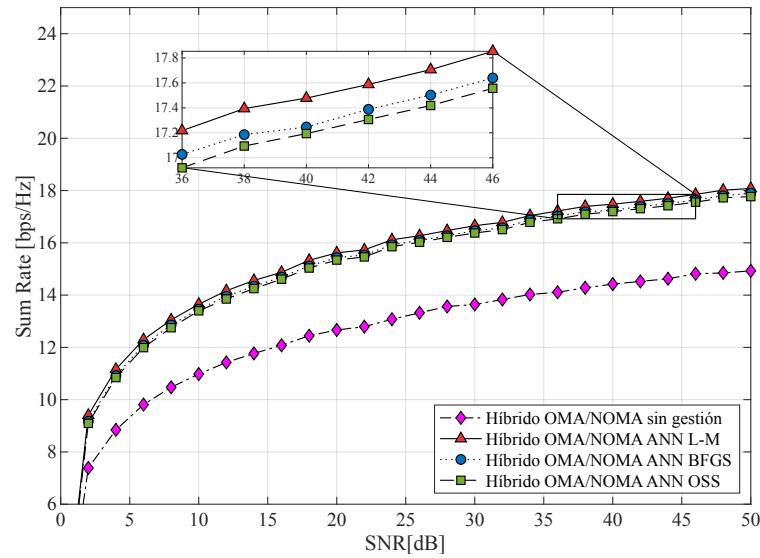


Figura 5.5: Rendimiento del sistema híbrido OMA/NOMA gestionado por una ANN con 50-30-20 neuronas en 3 HL.

5.3. Conclusiones

En este capítulo se ha evaluado el desempeño tanto de la ANN, como de la red móvil con el sistema híbrido OMA/NOMA una vez aplicado los resultados de la ANN. Para que la evaluación sea confiable, se han llevado a cabo una extensa cantidad de simulaciones con distintas arquitecturas, para los tres algoritmos de entrenamiento y dos funciones de activación de HL. Se ha presentado además, el análisis de la función de activación sigmoide y tangente hiperbólica aplicado a las HL. Se concluyó entonces, que la función de activación que presentó mejoras en precisión y tiempo de entrenamiento para la mayoría de algoritmos de entrenamiento fue la tangente hiperbólica.

Como era de esperarse, al aplicar un sistema híbrido OMA/NOMA con cualquiera de los algoritmos de entrenamiento, el desempeño de la red es mucho mejor que en el caso de un sistema híbrido gestionado aleatoriamente. Nuestra propuesta de usar una red neuronal artificial para la selección del método de acceso más adecuado para cada UE, incrementa la eficiencia de la red en términos de la tasa total del sistema en al menos un 20 % en comparación con una selección aleatoria del método de acceso.



Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones generales del presente trabajo. Adicionalmente, se ilustran algunas posibles mejoras que se pueden implementar y los trabajos futuros.

6.1. Conclusiones

Las redes de próxima generación tienen grandes desafíos por delante. Uno de ellos es la conectividad masiva usuarios, esto implica que cada usuario requiere de un espacio dentro del espectro de frecuencias para que pueda transmitir y recibir información. Por lo tanto, el espectro radioeléctrico debe ser gestionado de manera eficiente debido a que es un recurso escaso y costoso. En consecuencia es necesaria la implementación de mecanismos que administren eficientemente el método de acceso a este recurso.

El nuevo método de acceso **NOMA** en el dominio de la potencia es una técnica prometedora que permite ubicar a más de un usuario por cada subportadora. Este método en conjunto con el método tradicional de acceso (**OMA**) se presentan como la solución que mantiene un equilibrio entre el uso eficiente del espectro radioeléctrico y el rendimiento global del sistema.

En este trabajo, se propone la incorporación de una **ANN** a un sistema híbrido **OMA/NOMA** para maximizar el rendimiento de una red móvil con una conexión masiva. El objetivo principal es lograr un uso más eficiente del espectro de radio y una mejora en el rendimiento del sistema (*sum-rate*). En consecuencia, se evaluó el método de acceso híbrido gestionado a través de una **ANN** con diferentes algoritmos de entrenamiento y bajo diferentes arquitecturas. Con los resultados obtenidos de un extenso número simulaciones, se encontró que el algoritmo **L-M** presenta la mejor precisión para la mayoría de las arquitecturas evaluadas, muestra un incremento de hasta el 22,802 % en el rendimiento de la red. Por otro lado, el algoritmo **OSS** presenta el menor tiempo de entrenamiento en la mayoría de las arquitecturas **ANN**, pero con un bajo rendimiento de la red.

Para un entorno de red móvil de próxima generación, se puede concluir que el algoritmo **L-M** muestra los mejores resultados en el caso de requerir poco tiempo de entrenamiento (alrededor de 28 s) ya que en estas circunstancias exhibe mayor precisión que **BFGS** y **OSS**. En el caso de requerir una mayor precisión, **L-M** con 3 capas ocultas también evidencia mejores resultados a medida que se incrementa la cantidad de neuronas y de capas ocultas (22.802 %), a pesar de un tiempo de entrenamiento elevado (solo superado por **BFGS**). Sin embargo, es importante mencionar que esto aún se podría aplicar prácticamente en escenarios que admitan un entrenamiento fuera de línea (*offline*).

6.2. Trabajos Futuros

Como trabajo futuro, las **ANNs** se pueden evaluar adicionalmente con el error cuadrático medio y el descenso del gradiente, debido a que son métricas que también permiten evaluar las **ANNs**.



También se pretende incrementar la cantidad de [UEs](#) y se evaluarán escenarios más realistas, con diferentes modelos de canal que incluyan adicionalmente multi-trayecto. Además, se puede investigar la aplicación de estos resultados a una red masiva junto con distintas alternativas de métodos de agrupamiento. Adicionalmente, el trabajo podría complementarse agregando características ambientales, interferencia, calidad de servicio y otros requisitos específicos de los usuarios.



Apéndice A

Análisis Previo de Arquitecturas



A.1. Resultados del Análisis Previo de Arquitecturas de ANN

Resultados obtenidos de diferentes arquitecturas con diferentes algoritmos de entrenamiento.

Tabla A.1: Análisis previo con el algoritmo de entrenamiento **BFGS**

Número de Capas Ocultas	Neuronas por Capa Oculta	BFGS [trainbfg]			
		Iteraciones	Tiempo [mm:ss]	Error [%]	Validación
1	15	476	1:38	27.38 %	Validación
	30	537	2:47	26.00 %	Validación
	40	562	3:21	25.86 %	Validación
	50	886	6:51	25.04 %	Validación
	80	451	8:22	25.26 %	Validación
2	20-10	407	2:02	26.84 %	Validación
	30-10	609	4:52	25.32 %	Validación
	40-20	541	11:25	24.87 %	Validación
	50-30	459	34:36	24.74 %	Validación
	80-20	630	87:47	25.00 %	Validación
3	15-20-10	427	3:09	25.49 %	Validación
	25-35-25	322	22:53	25.35 %	Validación
	40-50-20	338	92:54	22.33 %	Validación
	60-80-30	290	937:32	24.99 %	Validación
	50-30-20	227	33:23	26.05 %	Validación
	40-20-10	564	17:08	25.23 %	Validación

**Tabla A.2:** Análisis previo con el algoritmo de entrenamiento **OSS**

Número de Capas Ocultas	Neuronas por Capa Oculta	One-Step Secant (OSS) [trainoss]			
		Iteraciones	Tiempo [mm:ss]	Error [%]	Detención
1	15	626	2:02	27.90 %	Validación
	30	890	3:48	26.56 %	Validación
	40	809	3:42	25.68 %	Validación
	50	723	4:06	26.23 %	Validación
	80	420	4:13	27.25 %	Validación
2	20-10	598	2:41	26.77 %	Validación
	30-10	664	4:14	26.32 %	Validación
	40-20	997	6:54	25.32 %	Max. Iter.
	50-30	617	7:08	26.42 %	Validación
	80-20	1000	15:27	25.81 %	Max. Iter.
3	15-20-10	338	2:08	26.76 %	Validación
	25-35-25	582	6:06	26.90 %	Validación
	40-50-20	318	5:39	27.66 %	Validación
	60-80-30	970	19:52	23.88 %	Validación
	50-30-20	419	5:55	25.68 %	Validación
	40-20-10	693	6:11	26.17 %	Validación

Tabla A.3: Análisis previo con el algoritmo de entrenamiento **L-M**

Número de Capas Ocultas	Neuronas por Capa Oculta	Levenberg-Marquardt (L-M) [trainlm]			
		Iteraciones	Tiempo [mm:ss]	Error [%]	Detención
1	15	46	00:21	27.20 %	Validación
	30	85	02:09	25.47 %	Validación
	40	70	02:42	25.10 %	Validación
	50	60	03:21	25.09 %	Validación
	80	66	15:07	24.66 %	Validación
2	20-10	103	02:19	23.88 %	Validación
	30-10	41	02:47	24.95 %	Validación
	40-20	88	20:46	22.86 %	Validación
	50-30	44	28:15	23.31 %	Validación
	80-20	42	64:32	22.76 %	Validación
3	15-20-10	99	04:51	25.00 %	Validación
	25-35-25	44	24:33	24.30 %	Validación
	40-20-10	36	11:19	23.47 %	Validación
	40-50-20	28	44:03	22.76 %	Validación
	50-30-20	25	27:35	22.89 %	Validación
	60-80-30	10	102:04	22.70 %	Validación



Apéndice B

Scripts



B.1. Script Generador de Datos

```
clear all %#ok<CLALL>
%% Parametros:

SNR_step=2;
Num_var=5000;          %Numero de elementos
UE_O=1;
UE_N=2;
SNR=[0:SNR_step:50];
%% SNR:

for j=1:length(SNR)
    %% Generaci n de Angulos:

    UE_angles=-90+rand(2,UE_N+UE_O)*180;
    for i=2:Num_var
        UE_angles(:, :, i)=-90+rand(2,UE_N+UE_O)*180;
    end
    save(strcat('Data/angles_SNR', num2str(SNR(j)), '.mat'), 'UE_angles')

    %Otra forma
    % Angles=[-90:5:90];
    % UE_angles2=nchoosek(Angles,3);
%% Generaci n del canal:

    CH_alpha=zeros(3,1,1);
    a=2;          %Exp de perdida LOS comun.usado = 2
    for i=1:Num_var
        CH_alpha(:, :, i)=(randn(3,1)+1i*randn(3,1))./sqrt(1+((0.5-0.01).*rand(3,1)+0.01).^2);
    end
    save(strcat('Data/channel_SNR', num2str(SNR(j)), '.mat'), 'CH_alpha')
end
```

B.2. Script Principal

```
clear all %#ok<CLALL>
%% Parametros:
% Para estasimulaci n se utiliza : 1 UE de OMA y 2 UE de NOMA

global freq UE_O lambda UE_N
SNR_step=2;
freq = 32e9;
lambda = physconst('LightSpeed')/freq;
```



```
UE_O=1;
UE_N=2;
SNR=[SNR_step:SNR_step:50]; %#ok<NBRAK>
%% C digo Principal:

for i=1:length(SNR)
    tic
    load(strcat('Data/angles_SNR',num2str(SNR(i)),'.mat')); ...
        %carga el archivo con UE_angles
    load(strcat('Data/channel_SNR',num2str(SNR(i)),'.mat'));
    Num_ele = size(UE_angles);
    for n=1:Num_ele(3)
        UE_anglesNComb=combnk({UE_angles(:,1,n),UE_angles(:,2,n),UE_angles(:,3,n)},2);
        AlphaComb=combnk(CH_alpha(:, :, n), 2);
        for j=1:3
            %NOMA: pre-processing
            UE_angle_N=cell2mat(UE_anglesNComb(j,:));
            ordNoma=sortrows([AlphaComb(j,:);UE_angle_N]','; %#ok<TRSRT> Ordena ...
                el array en funcion del ch solo noma

            UE_angle_N=ordNoma([2 3],:);
            BF_angle_N=mean(UE_angle_N,2);
            alpha_N1 = ordNoma(1,1);
            alpha_N2 = ordNoma(1,2);

            %OMA:
            UE_angle_O=UE_angles(:,4-j,n);
            BF_angle_O=UE_angle_O;
            alpha_O = CH_alpha(4-j, :, n);

            %Compute SumRate:
            [SR_URA_N,SR_URA_O]=URA(UE_angle_N,BF_angle_N,alpha_N1,alpha_N2, ...
                UE_angle_O,BF_angle_O,alpha_O,SNR(i));
            SumRate(j)= SR_URA_N+SR_URA_O;
        end
        Resul_SR(n,:)=SumRate;
        Resul_SR_MAX(n,:)= [find(SumRate==max(SumRate)),max(SumRate)];
    end
    toc
    % Guardar resultados
    save(strcat('Result/Resul_SR_SNR',num2str(SNR(i)),'.mat'),'Resul_SR')
    save(strcat('Result/Resul_SR_MAX_SNR',num2str(SNR(i)),'.mat'),'Resul_SR_MAX')
    disp(SNR(i))
end
```



Bibliografía

- [1] Ericsson, “Ericsson Mobility Report June 2019,” Tech. Rep., 2019. [En línea]. Disponible: www.ericsson.com/mobility-report
- [2] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, C.-L. I, y H. V. Poor, “Application of Non-Orthogonal Multiple Access in LTE and 5G Networks,” *IEEE Communications Magazine*, vol. 55, num. 2, pp. 185–191, feb 2017. [En línea]. Disponible: <http://ieeexplore.ieee.org/document/7842433/>
- [3] Y. Cai, Z. Qin, F. Cui, G. Y. Li, y J. A. McCann, “Modulation and Multiple Access for 5G Networks,” *IEEE Communications Surveys and Tutorials*, vol. 20, num. 1, pp. 629–646, 2018.
- [4] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, y V. K. Bhargava, “A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends,” *IEEE Journal on Selected Areas in Communications*, vol. 35, num. 10, pp. 2181–2195, 2017.
- [5] O. Kucur, G. Karabulut Kurt, M. Z. Shakir, y I. S. Ansari, “Nonorthogonal Multiple Access for 5G and beyond,” *Wireless Communications and Mobile Computing*, vol. 2018, num. 12, pp. 2347–2381, 2018.
- [6] L. Dai, B. Wang, Y. Yuan, S. Han, C. L. I, y Z. Wang, “Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends,” *IEEE Communications Magazine*, vol. 53, num. 9, pp. 74–81, 2015.
- [7] S. M. Islam, N. Avazov, O. A. Dobre, y K. S. Kwak, “Power-Domain Non-Orthogonal Multiple Access (NOMA) in 5G Systems: Potentials and Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 19, num. 2, pp. 721–742, 2017.
- [8] Z. Wu, K. Lu, C. Jiang, y X. Shao, “Comprehensive Study and Comparison on 5G NOMA Schemes,” *IEEE Access*, vol. 6, pp. 18 511–18 519, 2018.



- [9] Z. Ding, S. Member, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, V. K. Bhargava, y L. Fellow, “A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends,” *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 35, num. 10, 2017. [En línea]. Disponible: <http://www.ieee.org/publications{ }standards/publications/rights/index.html>
- [10] *Principios de comunicaciones móviles*. Ediciones de la UPC, S.L., 2003. [En línea]. Disponible: <https://books.google.com.ec/books?id=GAf5BKJOuWc>
- [11] “Orthogonal Frequency Division Multiple Access : Is it the Multiple Access System of the Future?” *AU-KBC Research Center, Anna University, India*, 2006. [En línea]. Disponible: <http://au-kbc.org/comm/Docs/Tutorials/OFDMA{ }BCW{ }cv6.pdf>
- [12] Z. Cao, U. Tureli, y Y.-D. Yao, “Orthogonal frequency division multiple access for broadband wireless systems: Algorithms and implementation,” Ph.D. dissertation, USA, 2004, aAI3127357.
- [13] K. Higuchi y A. Benjebbour, “Non-Orthogonal Multiple Access (NOMA) with successive interference cancellation for future radio access,” *IEICE Transactions on Communications*, vol. E98B, num. 3, pp. 403–414, mar 2015.
- [14] D. Duchemin, J. M. Gorce, y C. Goursaud, “Code Domain Non Orthogonal Multiple Access versus ALOHA: A simulation based study,” *2018 25th International Conference on Telecommunications, ICT 2018*, pp. 445–450, 2018.
- [15] P. A. Campos, “Beamformer Design and Performance in mmWave-NOMA Systems,” Master’s thesis, University of Southampton, 2019.
- [16] M. M. El-Sayed, A. S. Ibrahim, y M. M. Khairy, “Power allocation strategies for non-orthogonal multiple access,” in *2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)*, 2016, pp. 1–6.
- [17] M. Aldababsa, M. Toka, S. Gökçeli, G. Karabulut Kurt, y O. Kucur, “A Tutorial on Nonorthogonal Multiple Access for 5G and Beyond,” 2018. [En línea]. Disponible: <https://doi.org/10.1155/2018/9713450>
- [18] R. &. Schwarz, “Millimeter-Wave Beamforming: Antenna Array Design Choices & Characterization.” [En línea]. Disponible: <http://www.rohde-schwarz.com/appnote/1MA276>
- [19] W. Liu y S. Weiss, “Adaptive Wideband Beamforming,” in *Wideband Beamforming*. Chichester, UK: John Wiley & Sons, Ltd, mar 2010, pp. 19–60. [En línea]. Disponible: <http://doi.wiley.com/10.1002/9780470661178.ch2>



- [20] M. Reil y G. Lloyd, "Millimeter-Wave Beamforming: Antenna Array Design Choices and Characterization," *Rohde & Schwarz White paper 1MA276_2e*, p. 28, 2016. [En línea]. Disponible: <http://www.rohde-schwarz.com/appnote/1MA276>
- [21] B. Sklar, "Rayleigh fading channels in mobile digital communication systems .i. characterization," *IEEE Communications Magazine*, vol. 35, num. 7, pp. 90–100, 1997.
- [22] T. Lv, Y. Ma, J. Zeng, y P. T. Mathiopoulos, "Millimeter-Wave NOMA Transmission in Cellular M2M Communications for Internet of Things," *IEEE Internet of Things Journal*, vol. 5, num. 3, pp. 1989–2000, jun 2018.
- [23] L. Zhu, J. Zhang, Z. Xiao, X. Cao, D. O. Wu, y X. G. Xia, "Joint Power Control and Beamforming for Uplink Non-Orthogonal Multiple Access in 5G Millimeter-Wave Communications," *IEEE Transactions on Wireless Communications*, vol. 17, num. 9, pp. 6177–6189, sep 2018.
- [24] D. O. Hebb, *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall, 1949.
- [25] D. W. Patterson, "Artificial Neural Networks: Theory and Applications," p. 477, 1996.
- [26] P. Viñuela y I. León, *Redes de neuronas artificiales: un enfoque práctico*. Pearson Educación, 2004. [En línea]. Disponible: <https://books.google.com.ec/books?id=izJ6NQAAAJ>
- [27] D. Graupe, *Principles of Artificial Neural Networks*, ser. Advanced Series in Circuits and Systems. WORLD SCIENTIFIC, apr 2013, vol. 6, num. 9.
- [28] R. Battiti, "First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method," *Neural Computation*, vol. 4, num. 2, pp. 141–166, mar 1992.
- [29] J. E. Dennis y R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, jan 1996. [En línea]. Disponible: <http://www.siam.org/journals/ojsa.php>
- [30] H. Karim, S. R. Niakan, y R. Safdari, "Comparison of neural network training algorithms for classification of heart diseases," *IAES International Journal of Artificial Intelligence*, vol. 7, num. 4, pp. 185–189, 2018.
- [31] H. Hassannejad, M. Sirus, y R. Mehdizadeh, "Comparison and Evaluation of Artificial Neural Network (ANN) Training Algorithms in Predicting Soil Type Classification," *Bulletin of Environment, Pharmacology and Life Sciences*, vol. 4, num. 1, pp. 212–218, 2015. [En línea]. Disponible: <http://www.bepls.com>



- [32] Z. Cömert y A. Kocamaz, "A Study of Artificial Neural Network Training Algorithms for Classification of Cardiocography Signals," *Bitlis Eren University Journal of Science and Technology*, vol. 7, num. 2, pp. 93–103, 2017.
- [33] F. Dario Baptista, S. Rodrigues, y F. Morgado-Dias, "Performance comparison of ANN training algorithms for classification," *2013 IEEE 8th International Symposium on Intelligent Signal Processing, WISP 2013 - Proceedings*, pp. 115–120, 2013.
- [34] J. Pereda, J. De Lope, y D. Maravall, "Comparative analysis of artificial neural network training methods for inverse kinematics learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4177 LNAI, pp. 171–179, 2006.
- [35] M. T. Hagan y M. B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Transactions on Neural Networks*, vol. 5, num. 6, pp. 989–993, 1994.
- [36] Q. Mao, F. Hu, y Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 20, num. 4, pp. 2595–2621, 2018.
- [37] N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue, y K. Mizutani, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Communications*, vol. 24, num. 3, pp. 146–153, 2017.
- [38] G. Gui, H. Huang, Y. Song, y H. Sari, "Deep Learning for an Effective Nonorthogonal Multiple Access Scheme," *IEEE Transactions on Vehicular Technology*, vol. 67, num. 9, pp. 8440–8450, 2018.
- [39] Narengerile y J. Thompson, "Deep Learning for Signal Detection in Non-Orthogonal Multiple Access Wireless Systems," *2019 UK/China Emerging Technologies, UCET 2019*, pp. 1–4, 2019.
- [40] L. S. Ezema y C. I. Ani, "Artificial Neural Network Approach to Mobile Location Estimation in GSM Network," *International Journal of Electronics and Telecommunications*, vol. 63, num. 1, pp. 39–44, 2017.
- [41] J. Cui, Z. Ding, y P. Fan, "The Application of Machine Learning in mmWave-NOMA Systems," *IEEE Vehicular Technology Conference*, vol. 2018-June, pp. 1–6, 2018.
- [42] J. Cui, Z. Ding, P. Fan, y N. Al-Dhahir, "Unsupervised machine learning-based user clustering in millimeter-Wave-NOMA systems," *IEEE Transactions on Wireless Communications*, vol. 17, num. 11, pp. 7425–7440, 2018.
- [43] K. Janghel y S. Prakriya, "Performance of Adaptive OMA/Cooperative-NOMA Scheme With User Selection," *IEEE Communications Letters*, vol. 22, num. 10, pp. 2092–2095, 2018.



- [44] N. Nomikos, T. Charalambous, D. Vouyioukas, G. K. Karagiannidis, y R. Wichman, “Hybrid NOMA/OMA with Buffer-Aided Relay Selection in Cooperative Networks,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 13, num. 3, pp. 524–537, 2019.
- [45] M. Alkhawatrah, Y. Gong, G. Chen, S. Lambbotharan, y J. A. Chambers, “Buffer-aided relay selection for cooperative noma in the internet of things,” *IEEE Internet of Things Journal*, vol. 6, num. 3, pp. 5722–5731, 2019.
- [46] P. Xu, J. Quan, Z. Yang, G. Chen, y Z. Ding, “Performance Analysis of Buffer-Aided Hybrid NOMA/OMA in Cooperative Uplink System,” *IEEE Access*, vol. 7, pp. 168 759–168 773, 2019.
- [47] M. Hudson, B. Martin, T. Hagan, y H. B. Demuth, “Neural Network Toolbox™ 7 User’s Guide,” Tech. Rep., 1992. [En línea]. Disponible: www.mathworks.com
- [48] H. H. Örkücü y H. Bal, “Comparing performances of backpropagation and genetic algorithms in the data classification,” *Expert Systems with Applications*, vol. 38, num. 4, pp. 3703–3709, 2011.